7-2007

Semantics to Energize the Full Services Spectrum: Ontological Approach to Better Exploit Services at Technical and Business Levels

Amit P. Sheth
Wright State University - Main Campus, amit@sc.edu

LSDIS

Large Scale Distributed Information Systems

University of Georgia
Computer Science Department

Semantics to energize the full Services Spectrum:

Ontological approach to better exploit services at technical and business levels

Amit Sheth

LSDIS Lab, University of Georgia,
Athens, Georgia, UGA

Special Thanks: Kunal Verma

Challenges

"Each enterprise will measure and aspire to its own unique level of **dynamism** based on its individual purpose. It is about being nimble and **adaptable**. A fully integrated business platform can respond faster, and completely, to change. Whether it involves fulfilling a new mandate or embracing a new market opportunity. Some organizations will push the envelope, **automating event-triggered responses** for highly integrated closed-loop processes, setting the stage for **self-optimizing systems.**"

Sandra Rogers, White Paper: Business Forces Driving Adoption of Service Oriented Architecture, Sponsored by: SAP AG

Semantic Services Sciences (3S Model)

- Based on IBM's vision [1] of service sciences
  - Need to take a pervasive view of services
  - Modeling people and organizational aspects as well as technical aspects of services
- The 3S model [2]
  - Semantics for all types of services:

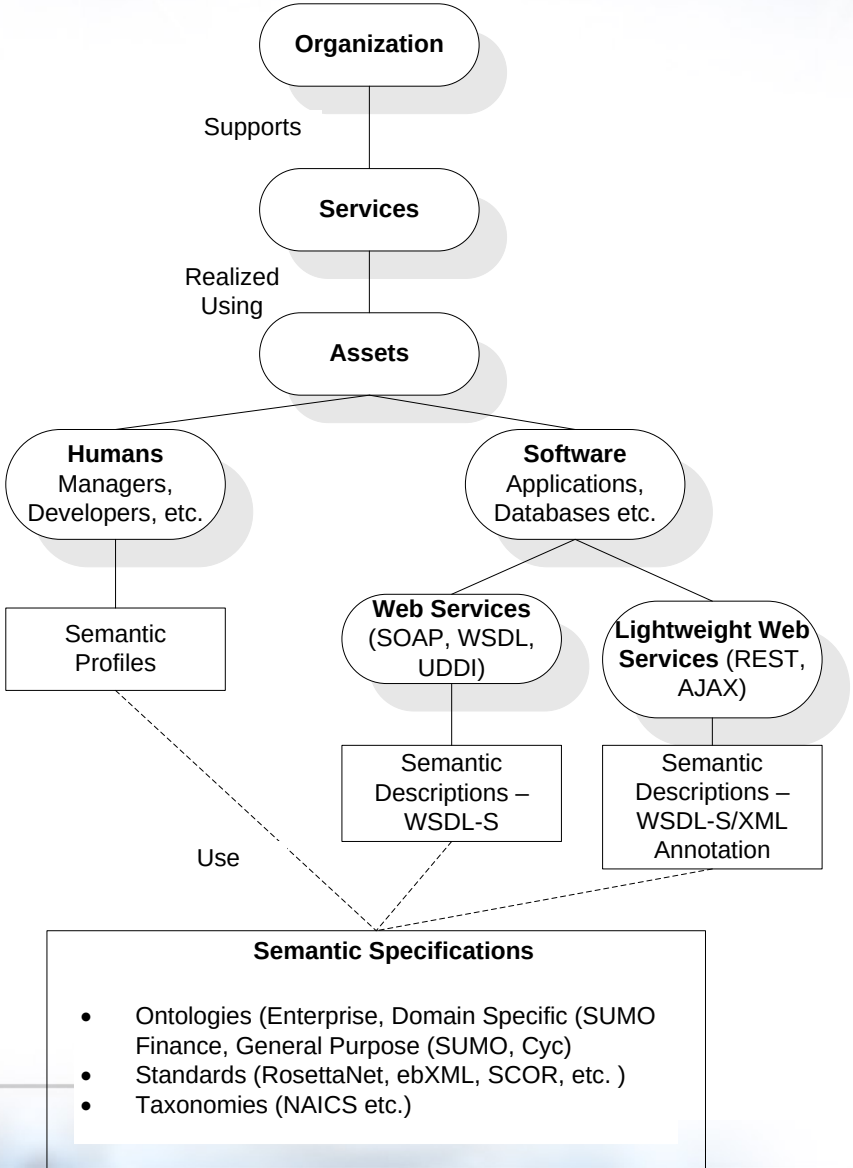
    Technical/Web Services to Knowledge Services

[1] IBM, Services Sciences, Management and Engineering, http://www.research.ibm.com/ssme/
[2] Amit P. Sheth, Kunal Verma, Karthik Gomadam, Semantics to energize the full Services Spectrum, Communications of the ACM (CACM) special issue on Services Science, July 2006

Using the 3S Model

- Consider global IT service provider developing a new multimedia service for UK telecom provider
  - Similar service already successfully provided in Japan
- To provide the new multimedia service
  - Business manager must leverage assets
  - Human assets
    - Teams in China (Telco Equipment), India (Telco SW, Back Office)
    - People who have domain expertise in the new market
    - Project Management, …
  - Technical assets
    - Reuse SW assets and compose services to create technical platform
    - Use lightweight services for information aggregation and GUIs

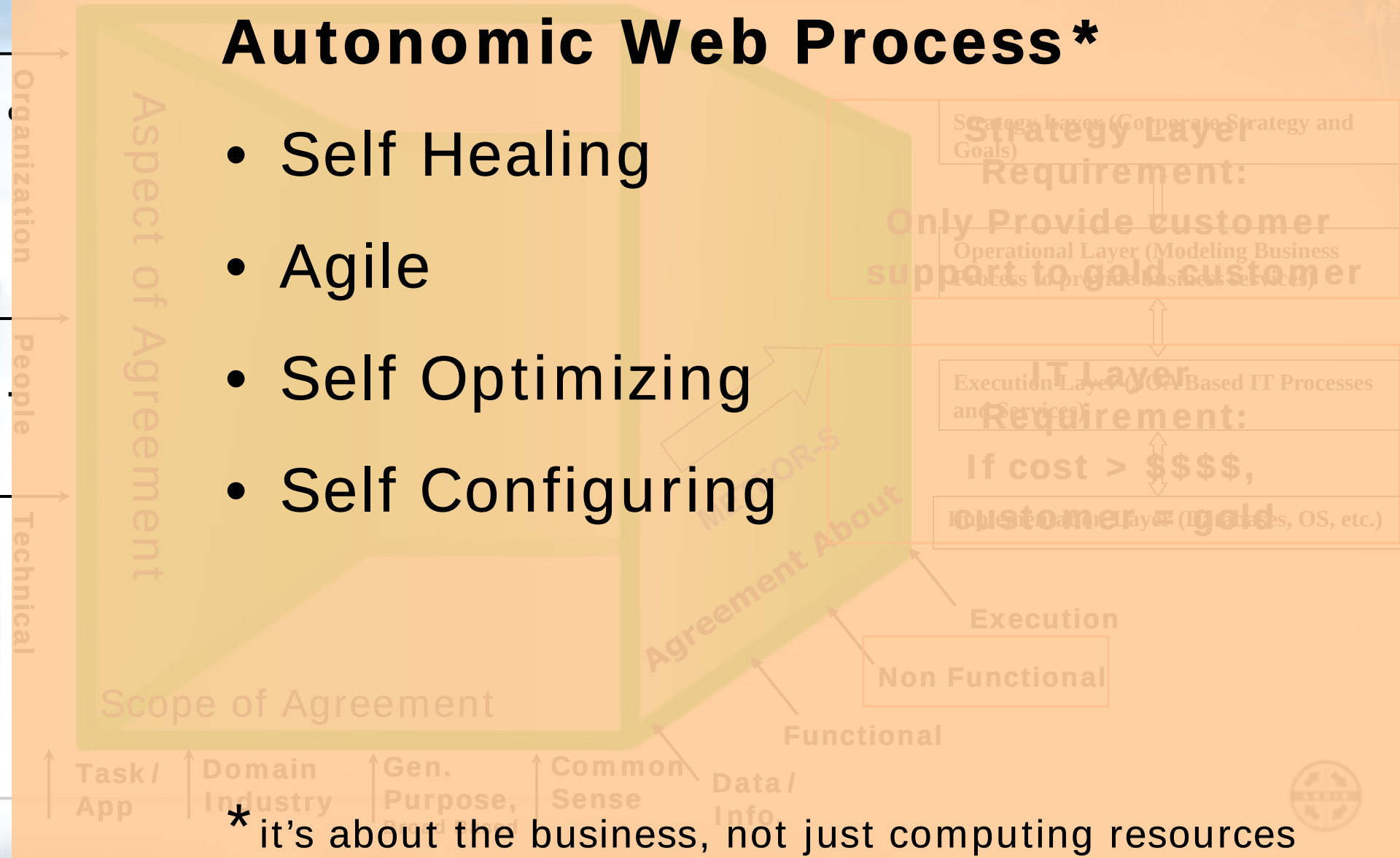
Semantic Services Sciences (3S Model)



Ontologies to Describe Service Semantics
(ontologies are about agreements)

Autonomic Web Process*

- Self Healing
- Agile
- Self Optimizing
- Self Configuring



* it's about the business, not just computing resources

Outline

- Semantics for Technical Services
  - Data Semantics *
  - Functional Semantics *
  - Non Functional Semantics *
  - Execution Semantics
- Semantics for Knowledge Services
- Conclusions

***Can be represented using ontologies**

LSDIS

Large Scale Distributed Information Systems

University of Georgia
Computer Science Department

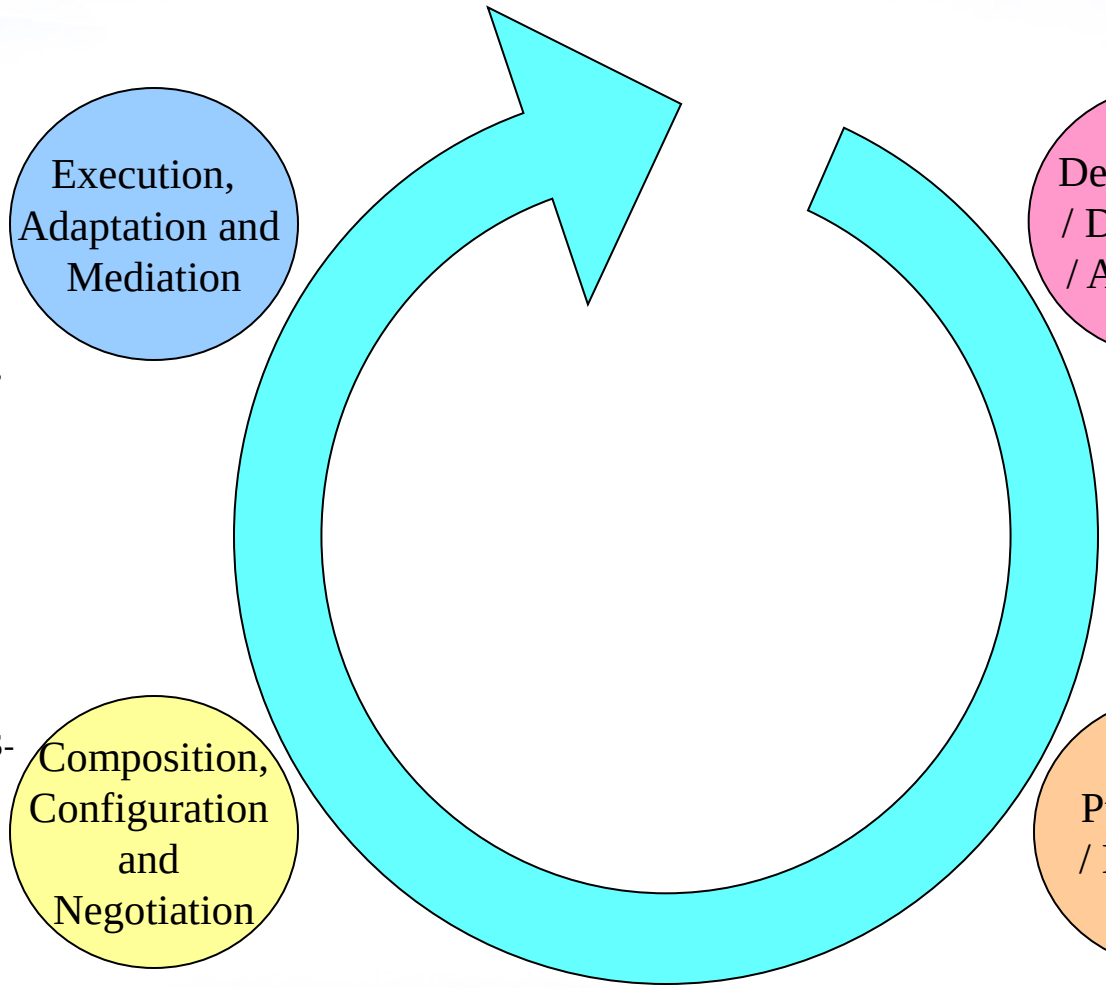
Semantics for Technical Services

Current and past focus of METEOR-S

Semantics for Technical Services

- Data/Information Semantics
  - **What:** (Semi-)Formal definition of data in input and output messages of a web service
  - **Why:** for discovery and interoperability
  - **How:** by annotating *input/output data* of web services using ontologies
- Functional Semantics
  - (Semi-) Formally representing capabilities of web service
  - for discovery and composition of Web Services
  - by annotating *operations* of Web Services as well as provide *preconditions* and *effects*
- Execution Semantics
  - (Semi-) Formally representing the execution or flow of a services in a process or operations in a service
  - for analysis (verification), validation (simulation) and execution (exception handling) of the process models
  - using *State Machines*, *Petri nets*, *activity diagrams* etc.
- Non Functional Semantics (WS-*)
  - (Semi-) formally represent qualitative and quantitative measures of Web process
  - Non- Quantitative includes security, transactions
  - Quantitative includes cost, time etc.
  - Business constraints and inter service dependencies (Domain and application ontologies)

Semantics for Technical Services



BPWS4J,

activeBPEL, WSMX

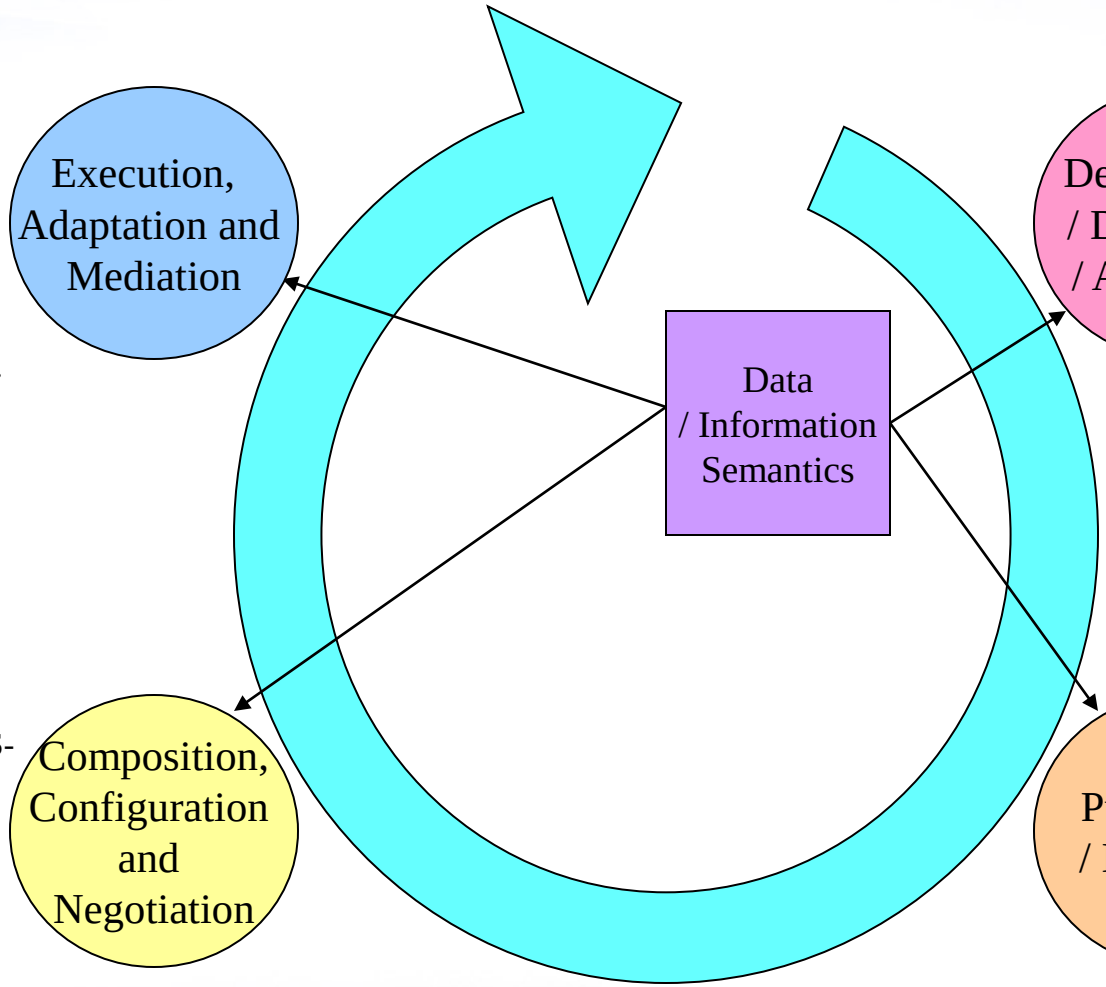
METEOR-S

WSDL, WSDL-S, SAWSDL, WSMO, OWL-S
METEOR-S
(MWSAF)

BPEL, WS-Agreement, WS-Policy
METEOR-S
(MWSCF)

(Semantic) UDDI
METEOR-S
(MWSDI)

Semantics for Technical Services



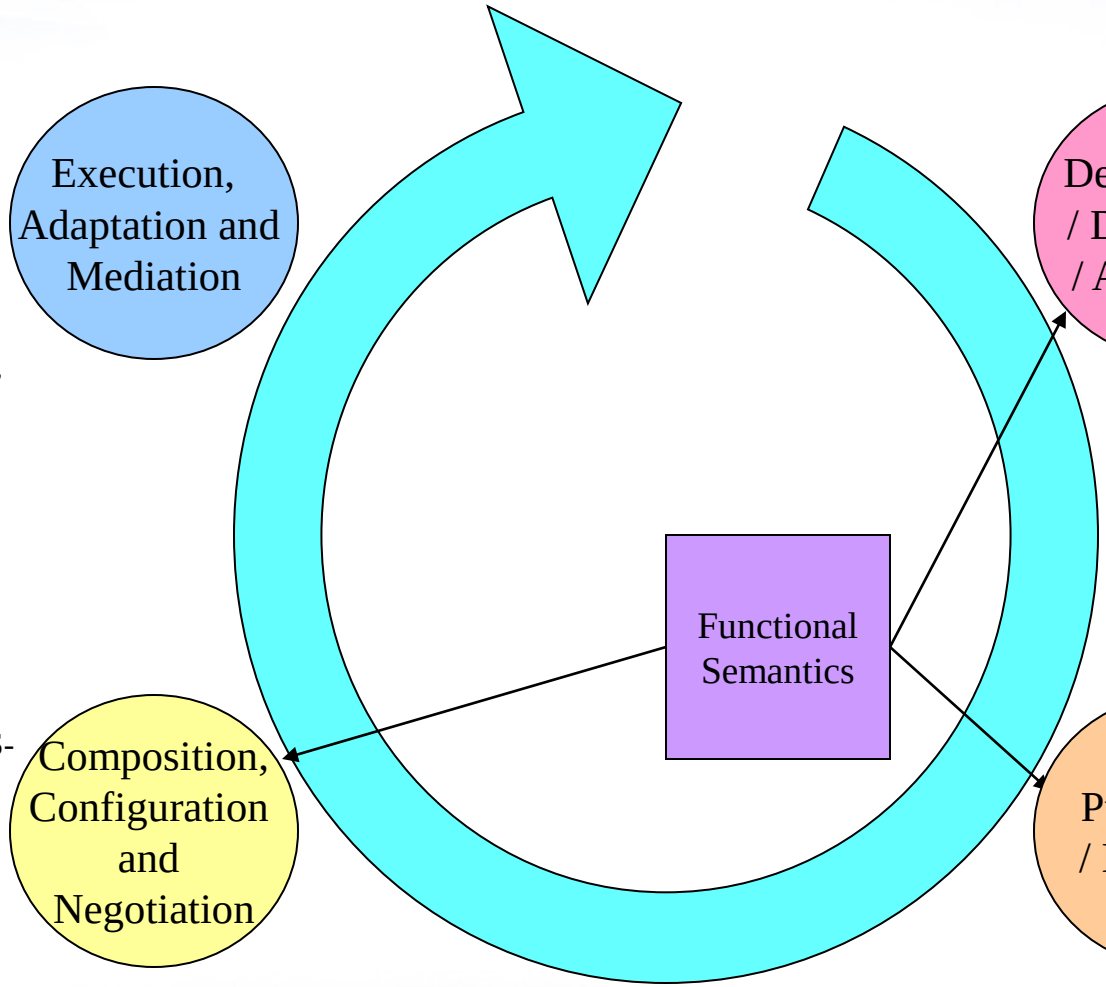
BPWS4J,
activeBPEL.
WSMX
METEOR-S

WSDL, WSDL-S, SAWSDL, WSMO, OWL-S
METEOR-S
(MWSAF)

BPEL, WS-Agreement, WS-Policy
METEOR-S
(MWSCF)

(Semantic) UDDI
METEOR-S
(MWSDI)

Semantics for Technical Services



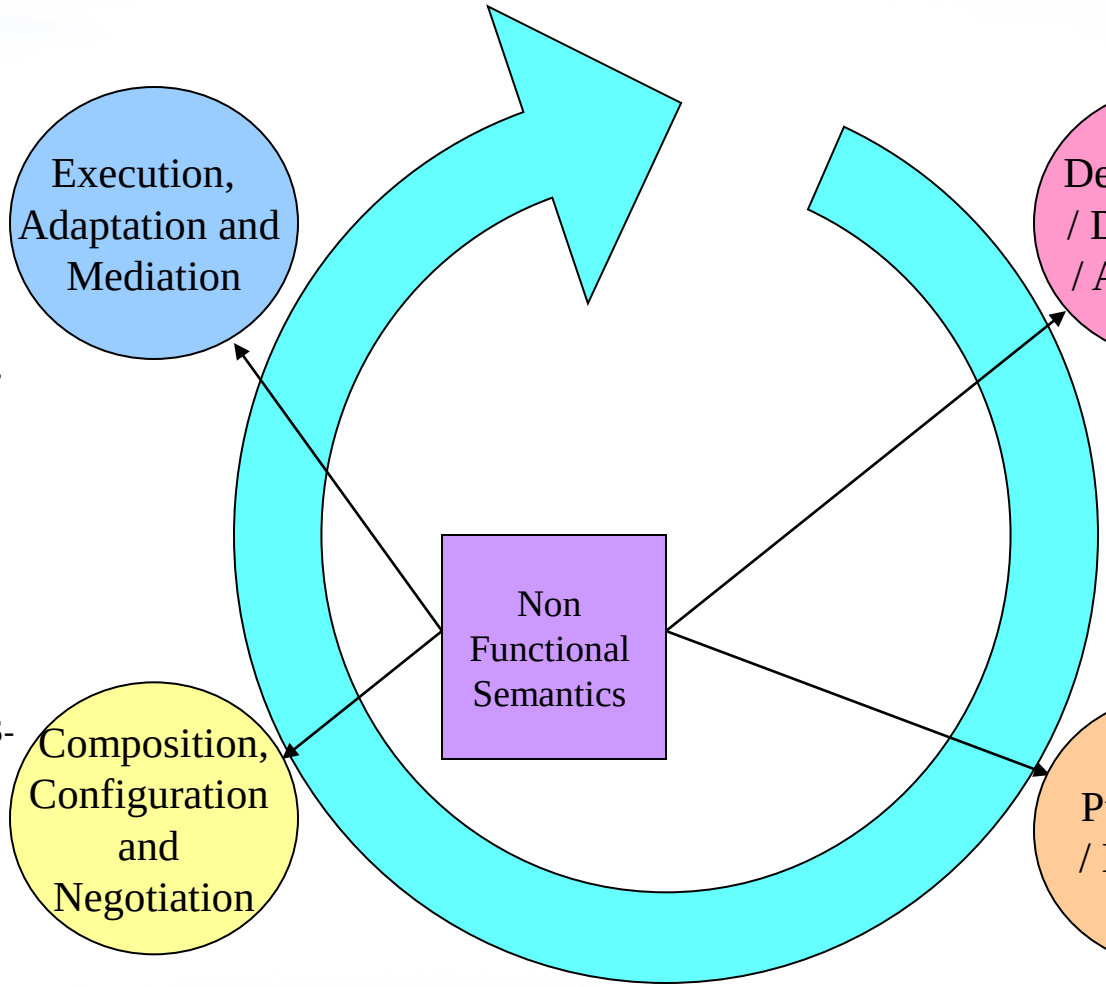
BPWS4J,
activeBPEL,
WSMX
METEOR-S

WSDL, WSDL-S, SAWSDL, WSMO, OWL-S
METEOR-S
(MWSAF)

BPEL, WS-Agreement, WS-Policy
METEOR-S
(MWSCF)

(Semantic) UDDI
METEOR-S
(MWSDI)

Semantics for Technical Services



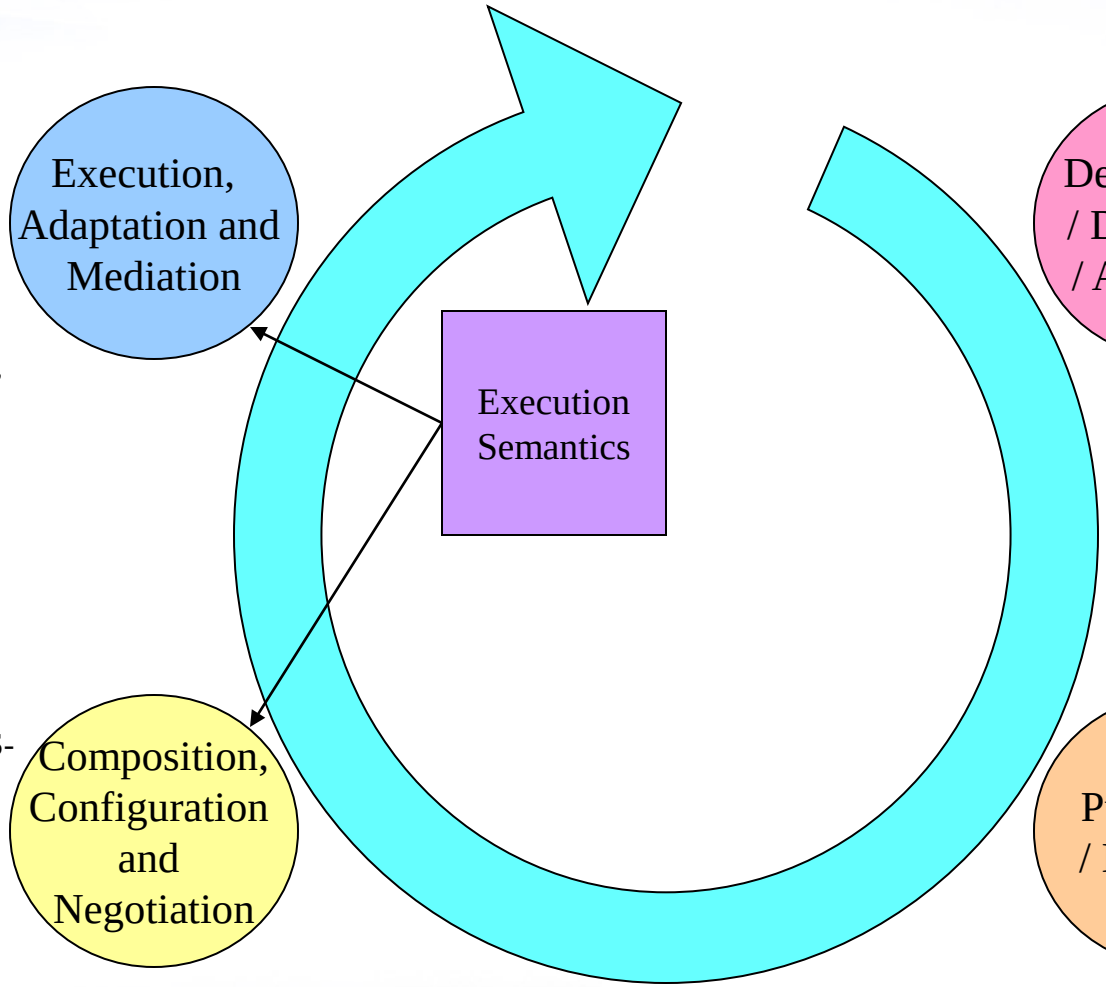
BPWS4J,
activeBPEL,
WSMX
METEOR-S

WSDL, WSDL-S, SAWSDL, WSMO, OWL-S
METEOR-S
(MWSAF)

BPEL, WS-Agreement, WS-Policy
METEOR-S
(MWSCF)

(Semantic) UDDI
METEOR-S
(MWSDI)

Semantics for Technical Services



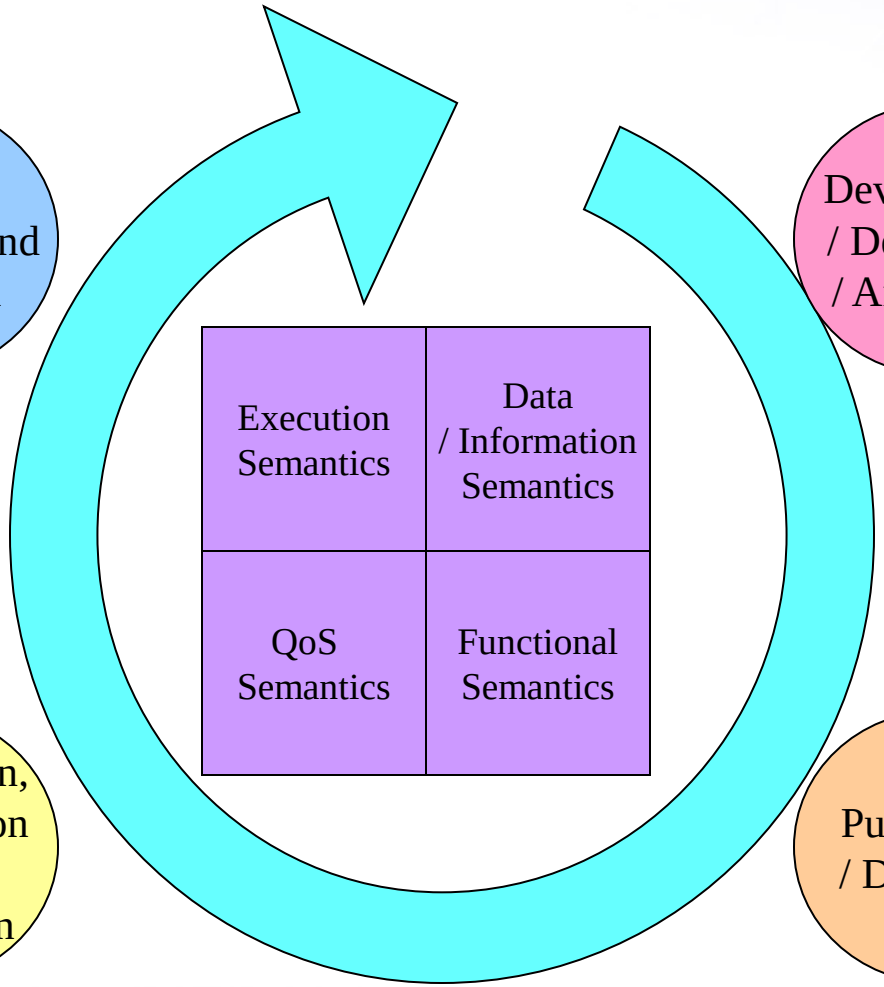
BPWS4J,
activeBPEL,
WSMX
METEOR-S

WSDL, WSDL-S, SAWSDL, WSMO, OWL-S
METEOR-S
(MWSAF)

BPEL, WS-Agreement, WS-Policy
METEOR-S
(MWSCF)

(Semantic) UDDI
METEOR-S
(MWSDI)

Semantics for Technical Services



BPWS4J,
activeBPEL,
WSMX
METEOR-S

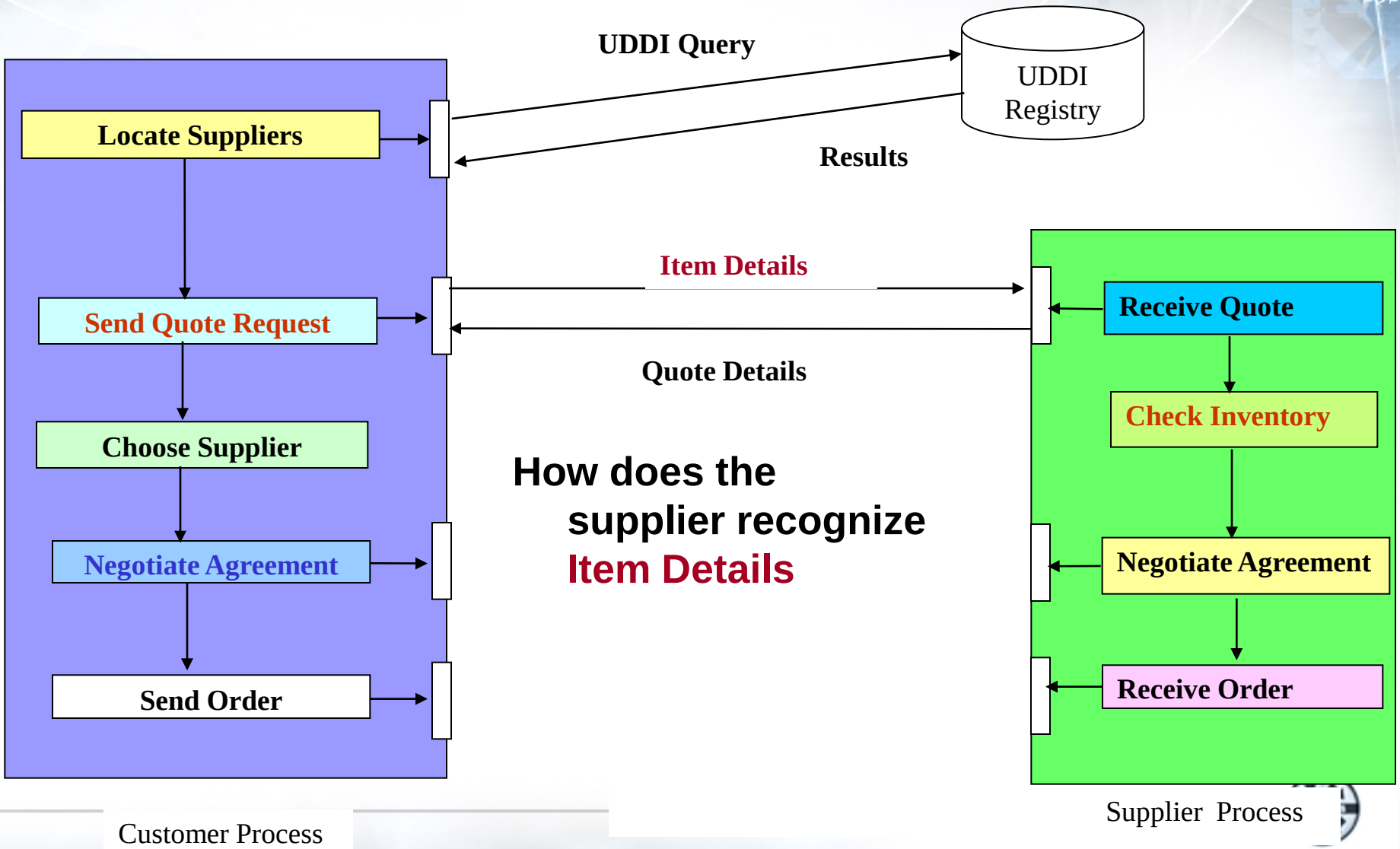
WSDL, WSDL-S, SAWSDL, WSMO, OWL-S
METEOR-S
(MWSAF)

BPEL, WS-Agreement, WS-Policy
METEOR-S
(MWSCF)

(Semantic) UDDI
METEOR-S
(MWSDI)

DATA SEMANTICS

Data Semantics



Data Semantics - options

- Pre-defined agreement on all data fields
  - Limited flexibility, hard to integrate new suppliers in process
- Use a standard like Rosetta Net/ebXML
  - Greater flexibility, but limited to suppliers following standard
  - Standard may not be expressive enough for everyone's needs
- Annotate data fields with domain ontologies
  - Most flexible, semi-automatic transformation based on ontology mapping
  - Ontology can be based on domain standard, while providing more flexibility and extensibility

LSDIS

Large Scale Distributed Information Systems

University of Georgia
Computer Science Department

WSDL-S Specification

(Now the key input to W3C leading to Semantic Annotation of WSDL-SAWSDL)

PurchaseOrder.wsdls

.....

<xs:element name= " **OrderConfirmation**" type="xs:string

wssem:modelReference="**rosetta#PurchaseOrderResponse**"/>

</xs:schema>

</types>

<interface name="PurchaseOrder">

<**wssem:category** name= “Electronics” taxonomyURI=http://www.naics.com/
taxonomyCode=”443112” />

<operation name=“**order**” pattern=wsdl:in-out
modelReference = "**rosetta#RequestPurchaseOrder**" >

<input messageLabel = ”processPurchaseOrderRequest"
element="tns:processPurchaseOrderRequest"/>

<output messageLabel ="processPurchaseOrderResponse"
element="processPurchaseOrderResponse"/>

<!—Precondition and effect are added as extensible elements on an operation>

<**wssem:precondition** name="ExistingAcctPrecond"

wssem:modelReference="POOntology#AccountExists">

<**wssem:effect** name="ItemReservedEffect"

wssem:modelReference="POOntology#ItemReserved"/>

</operation>

</interface>

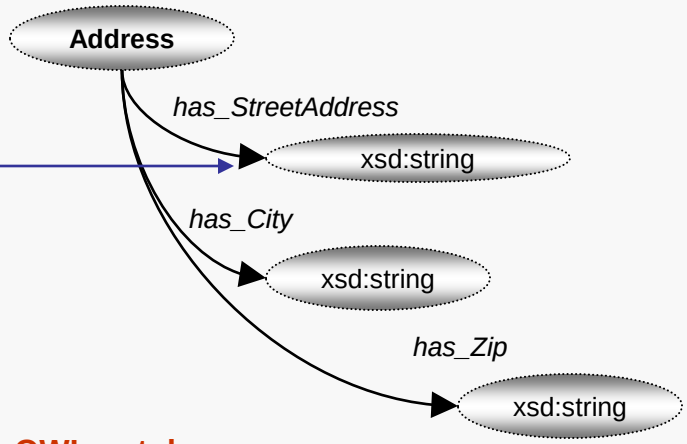
Data from Rosetta Net Ontology

Function from Rosetta Net Ontology

Representing mappings

```
<complexType name="POAddress"
wssem:schemaMapping="http://www.ibm.com/schemaMapping/POAddress.xsl#input-doc=doc("POAddress.xml")">
<all>
<element name="streetAddr1" type="string" />
<element name="streetAdd2" type="string" />
<element name="poBox" type="string" />
<element name="city" type="string" />
<element name="zipCode" type="string" />
<element name="state" type="string" />
<element name="country" type="string" />
<element name="recipientInstName" type="string" />
</all>
</complexType>
```

WSDL complex type element



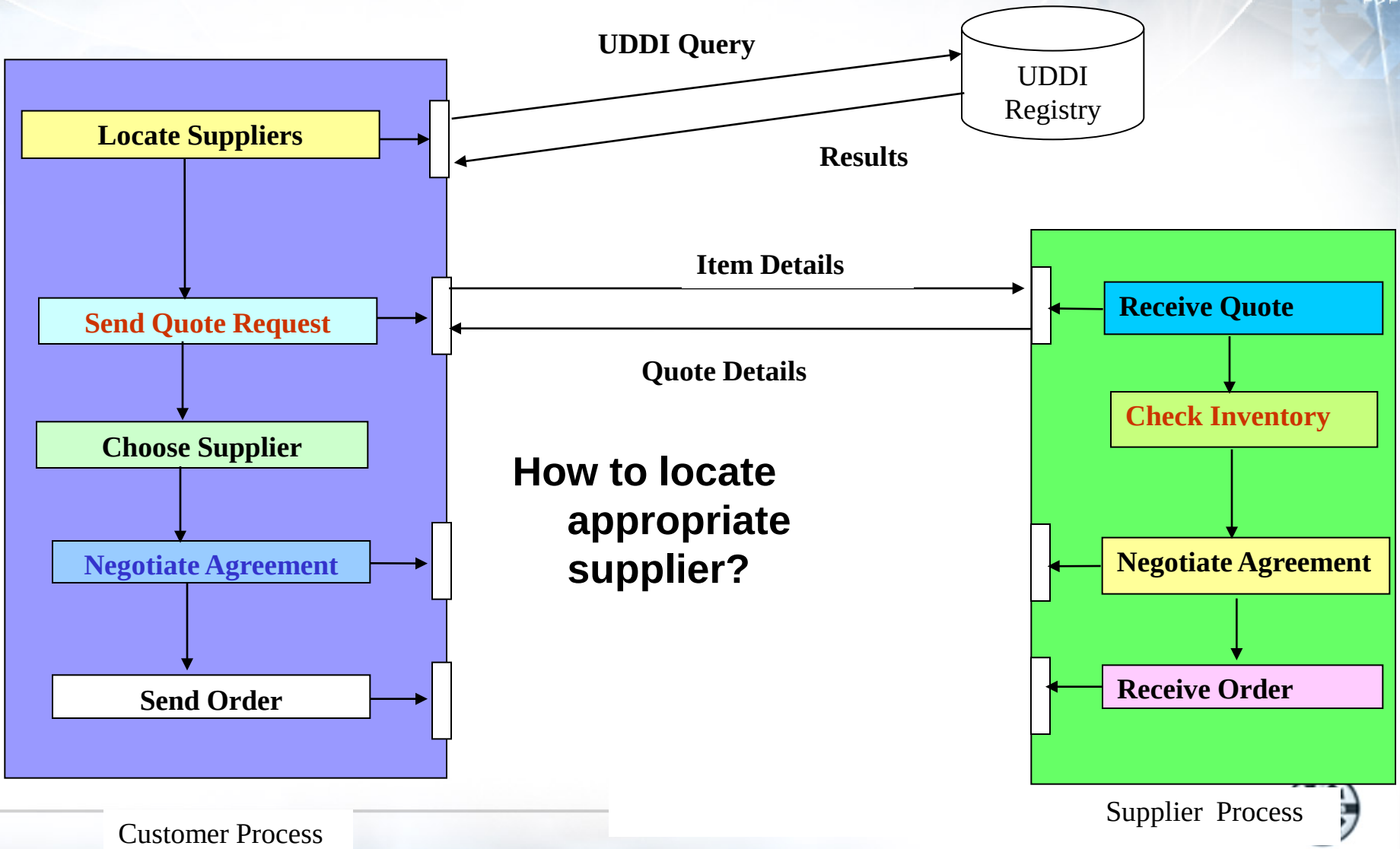
OWL ontology

Mapping using XSLT

```
....
<xsl:template match="/">
<POOntology:Address rdf:ID="Address1">
<POOntology:has_StreetAddress rdf:datatype="xs:string">
<xsl:value-of select="concat(POAddress/streetAddr1,POAddress/streetAddr2)"/>
</POOntology:has_StreetAddress >
<POOntology:has_City rdf:datatype="xs:string">
<xsl:value-of select="POAddress/city"/>
</POOntology:has_City>
<POOntology:has_State rdf:datatype="xs:string">
<xsl:value-of select="POAddress/state"/>
</POOntology:has_State>....
```

FUNCTIONAL SEMANTICS

Functional Semantics



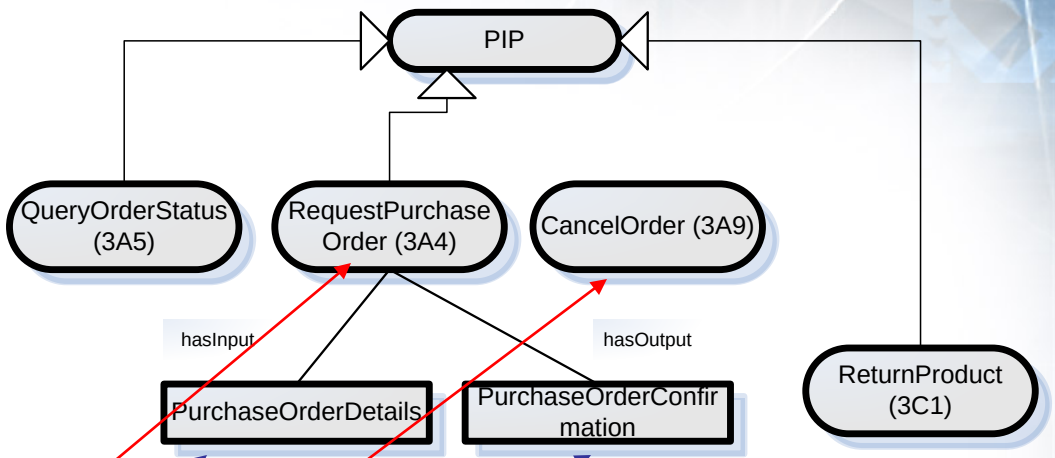
Functional Semantics

- Keyword based search in UDDI
  - Needs human involvement
  - Low precision and high recall
- Port Type based search in UDDI
  - Requires service providers to agree on port types
  - Less flexible, requires total agreement on method names and data type names
- Template Based Semantic Discovery
  - Requires ontological commitment of data types and operations
  - Can search on any or many aspects of description+interface
  - Can have complex similarity measures and be used to provide ranked results based on similarity

Semantic Templates

- Semantic Templates capture the functionality of a Web service with the help of ontologies/other domain models
- Find a service that sells RAM in Athens, GA. It must allow the user to return and cancel, if needed
- The template can also have non-functional (QoS) requirements such as response time, security, etc.

Part of Rosetta Net Ontology



SEMANTIC TEMPLATE

Service Level Metadata (SLM)
IndustryCategory = NAICS:Electronics
ProductCategory = DUNS:RAM
Location = Athens, GA

Operation 1
Action = Rosetta#RequestPurchaseOrder
Input = Rosetta#PurchaseOrderRequest
Output = Rosetta#PurchaseConfirmation
Policy = {Encryption = RSA, ResponseTime < 5 sec}

Operation 2
Action = Rosetta#CancelOrder
……..

WSDL-S is used to capture semantic templates

Data Semantics
Functional Semantics
Non-Functional Semantics

Semantic Discovery

- Finds actual services matching semantic templates
- Implemented as a layer over UDDI [1]
- Current implementation based on ontological representation of operations, inputs and outputs
- Returns ranked of services for each semantic template
- Builds upon following previous discovery implementations
  - Extends matching presented in [2] to consider operations and service level metadata
  - Extends the approach presented “WSDL to UDDI Mapping” [3] to support operation level discovery

[1] K. Verma, K. Sivashanmugam, A. Sheth, A. Patil, S. Oundhakar and John Miller, METEOR-S WSDI: A Scalable Infrastructure of Registries for Semantic Publication and Discovery of Web Services, JITM

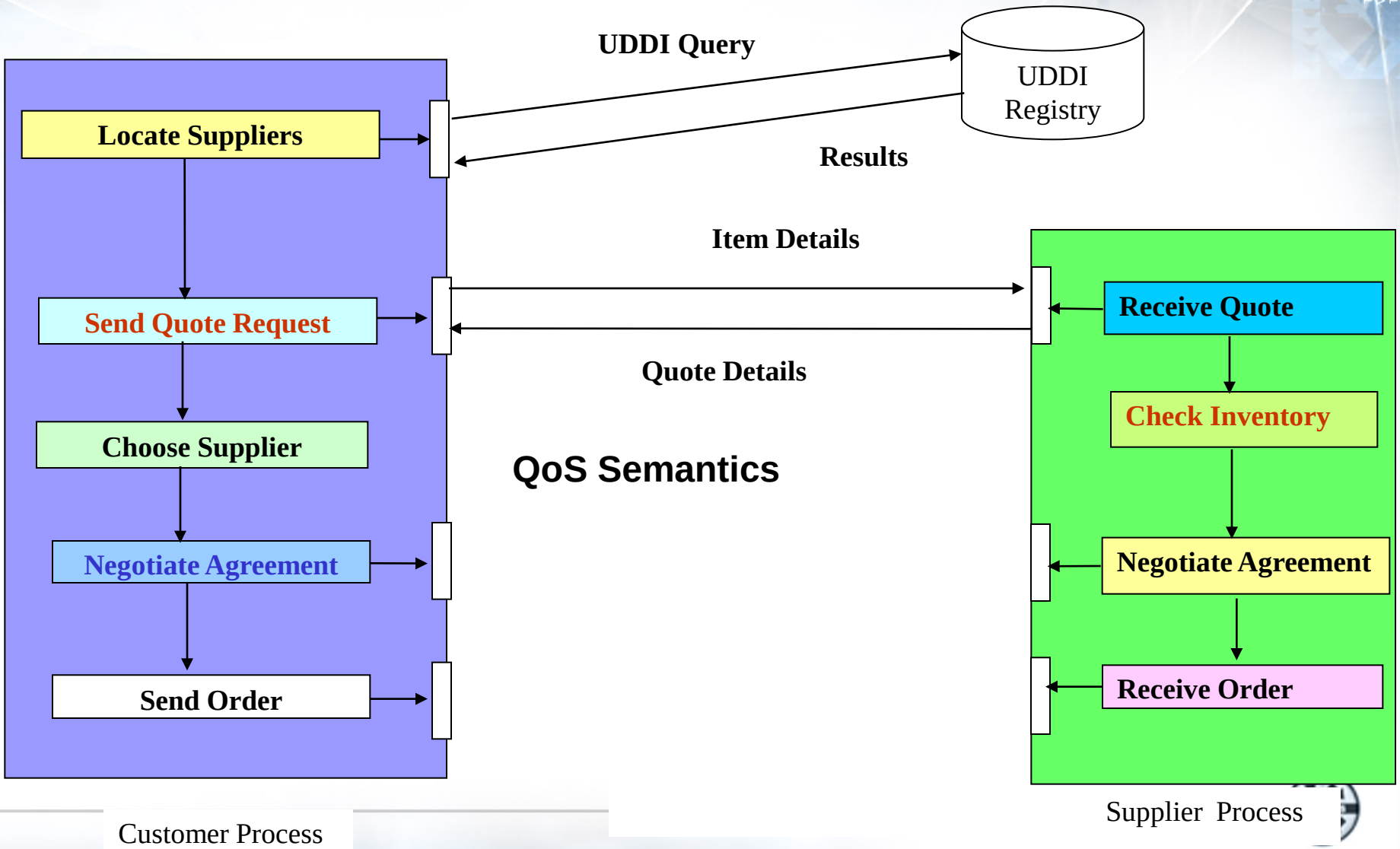
[2] M. Paolucci, T. Kawamura, T. Payne and K. Sycara, Semantic Matching of Web Services Capabilities, ISWC 2002.2

[3] Using WSDL in a UDDI Registry, Version 2.0.2 - Technical Note, http://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-wsdl-v202-20040631.pdf

Non Functional Semantics

Business and Application constraints

Non Functional Semantics



Non Functional Semantics

- Does the supplier support customer's business constraints
  - e.g. cost, supply time etc.
- Interaction should adhere to the entities' policies
  - e.g security, transactions
- In case of more suppliers, domain constraints should be satisfied
  - e.g. a certain supplier's parts do not work with other supplier's parts

Non Functional Semantics

- Used in lifecycle
  - Agreement Matching
    - Matching syntactically heterogeneous by semantically homogeneous agreements
  - Dynamic Process Configuration
    - Configuring process based on process constraint

We will demonstrate how ontology-driven semantic approach supports these capabilities.

SWAPS: Use of Semantics in Agreement Matching

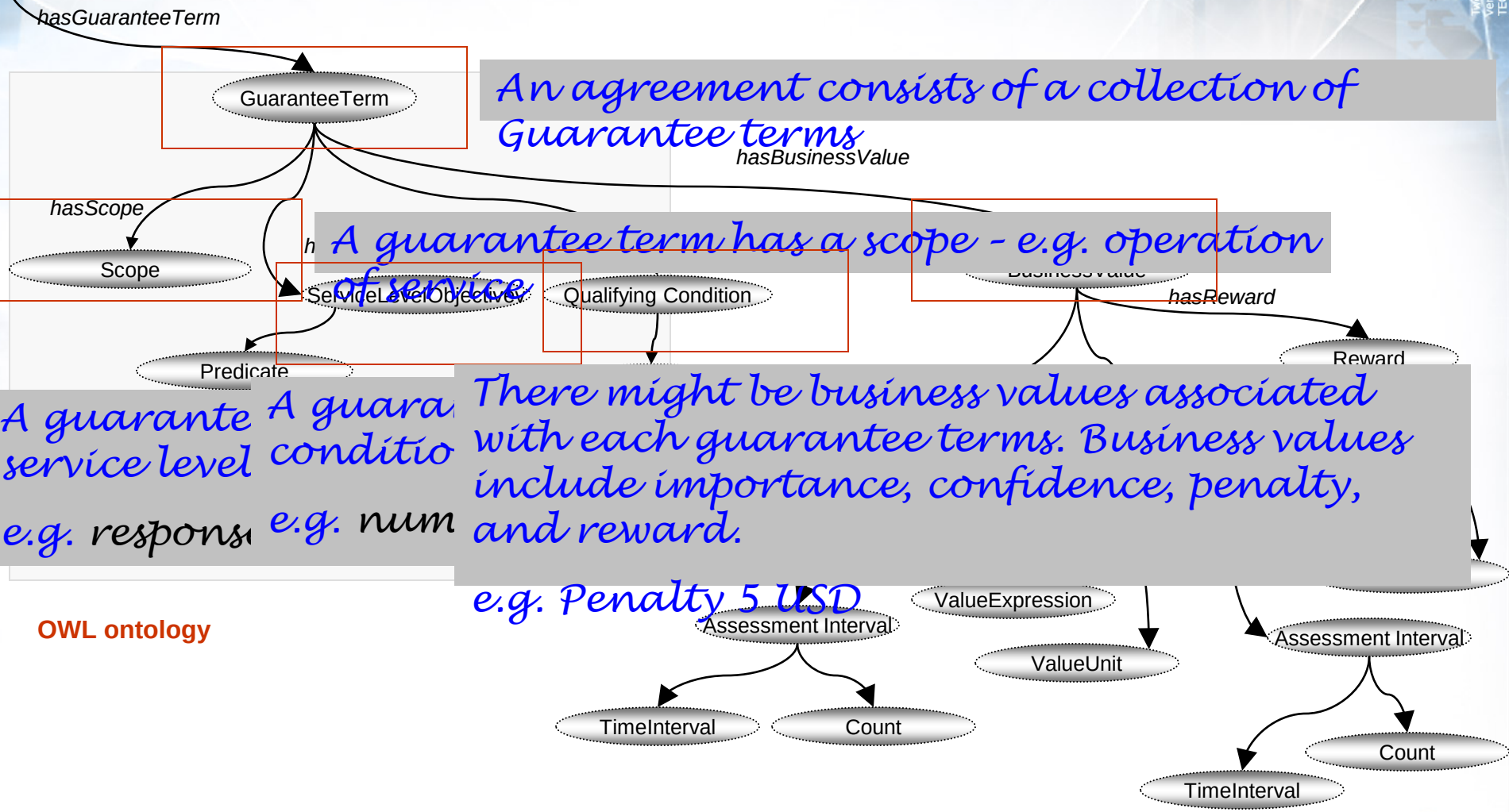
An agreement is a collection of alternatives.
A={Alt1, Alt2, …, AltN}

An alternative is a collection of guarantees.
Alt={G1, G2, …GN}

*"**requirement(Alt, G)**" returns true if G is a requirement of Alt*
*"**capability(Alt, G)**" returns true if G is an assurance of Alt*
*"**scope(G)**" returns the scope of G*
*"**obligation(G)**" returns the obligated party of G*
*"**satisfies(Gj, Gi)**" returns true if the SLO of Gj is equivalent to or stronger than the SLO of Gi*

An alternative Alt1 is a suitable match for Alt2 if:
(∀Gi) such that Gi ∈ Alt1 ∧ requirement(Alt1, Gi) ∧ (∃Gj) such that Gj ∈ Alt2 ∧ capability(Alt2, Gj) ∧ scope(Gi) = scope(Gj) ∧ obligation(Gi) = obligation(Gj) ∧ satisfies(Gj, Gi)

WS-Agreement Definition and Ontology



An agreement consists of a collection of Guarantee terms

A guarantee term has a scope - e.g. operation of service

A guarante
service level
e.g. respons

A guara
conditio
e.g. num

There might be business values associated with each guarantee terms. Business values include importance, confidence, penalty, and reward.

e.g. Penalty 5 USD

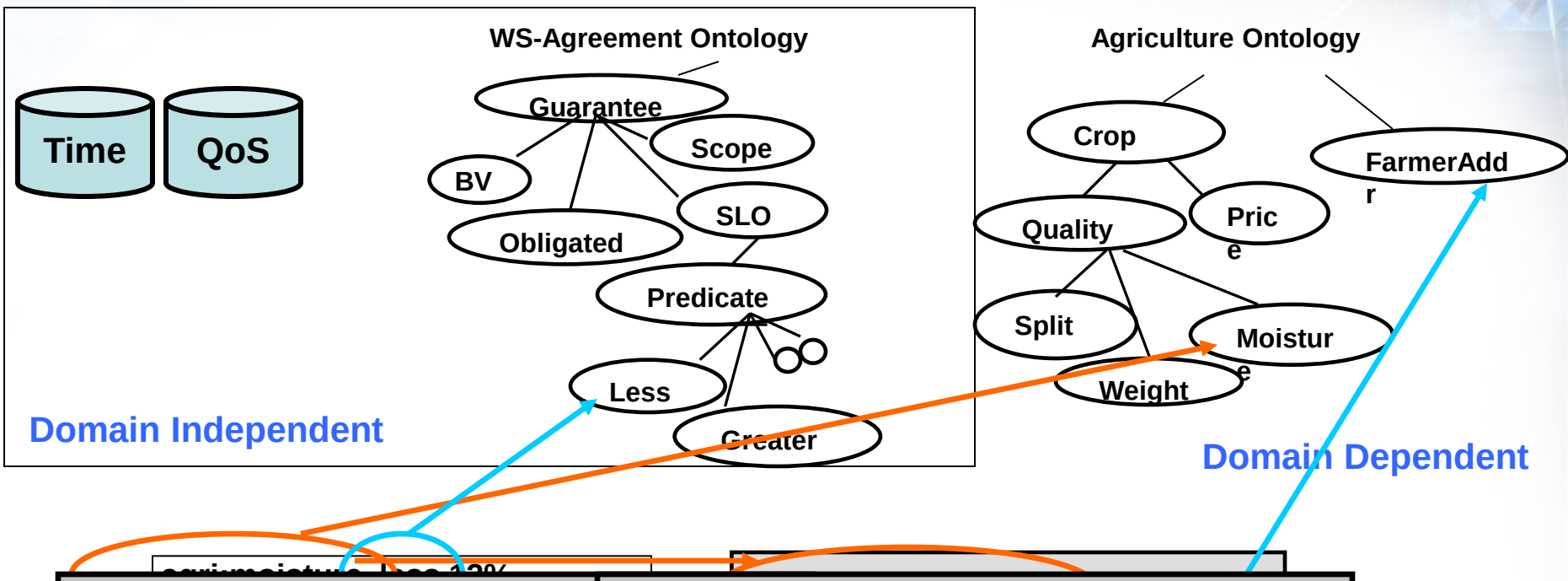
OWL ontology

Agreement represented as an instance of ontolog

SWAPS Ontologies

- **WS-Agreement:** individual agreements are instances of the WS-Agreement ontology
- **Temporal Concepts:** time.owl (OWL version of DAML time http://www.isi.edu/~pan/damltime/time.owl)
  - Concepts: seconds, dayOfWeek, ends
- **Quality of Service:** Max Maximilien's QoS ontology (IBM) -> Ont-Qos
  - Concepts: responseTime, failurePerDay
- **Domain Ontology:** an ontology used to represent the domain

Using Semantic Agreements with WSDL-S

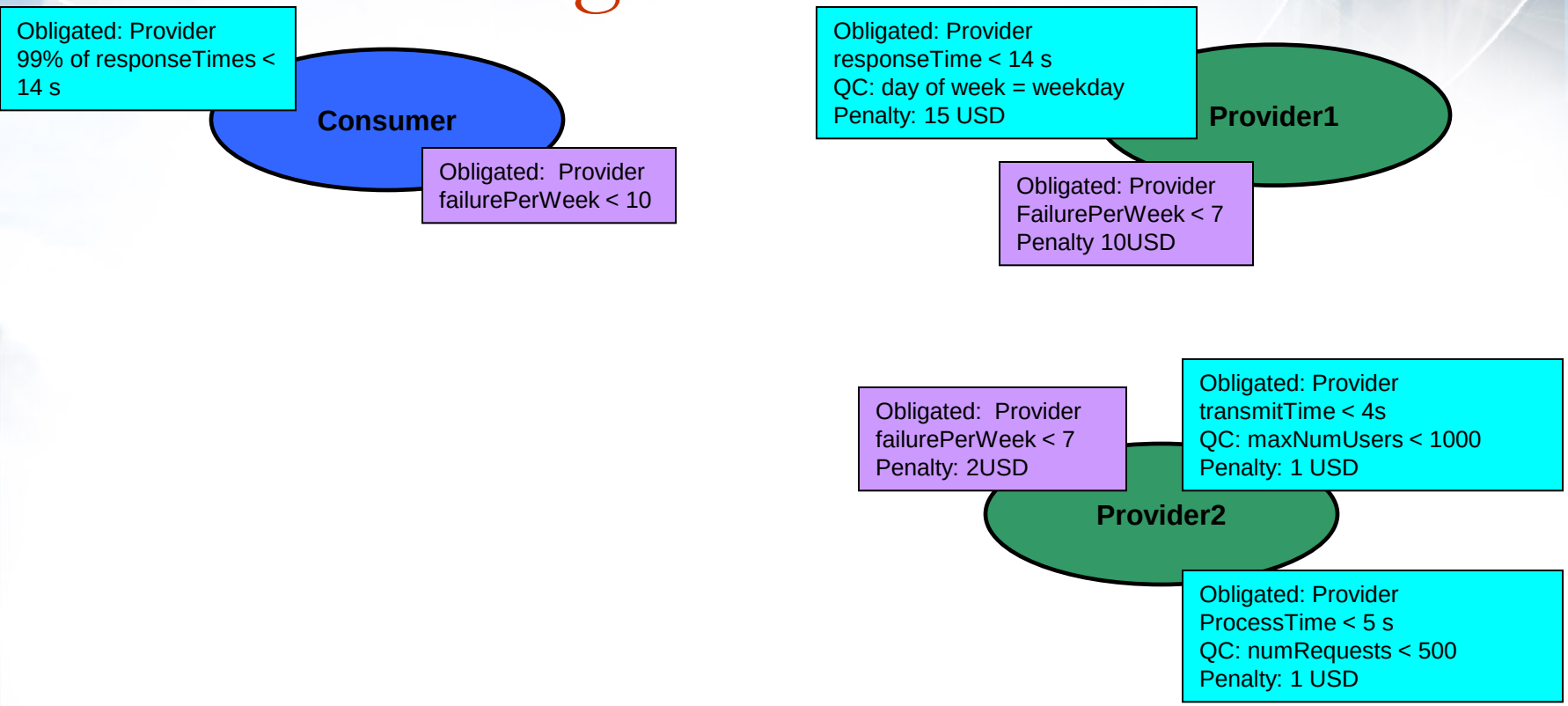


Adding Semantics to Agree	Adding Semantics to Web Services:
Improves Monitoring and N	Enables more accurate discovery and composition.
Improves the accuracy of m	

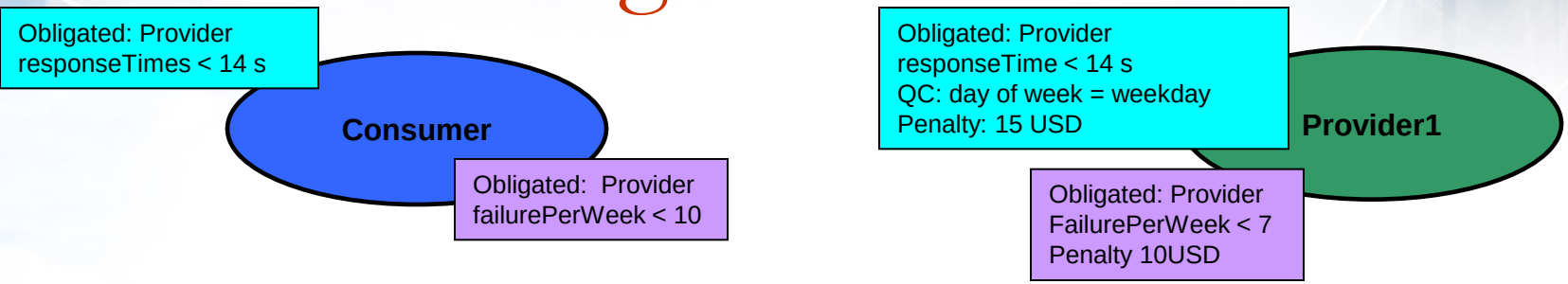
Evaluation

Consumer Requirement	Provider Capability	Approach 1: Ontology and Rules	Approach 2: Ontology without Rules	Approach 3: Rules without Ontologies	Approach 4: No Rules and No Ontology
responseTime < 5	responseTime < 4	YES	YES	YES, but only if parameters are named similar syntactically	YES, but only if parameters are named similar syntactically
responseTime < 5	(duration1 + duration2) < 4	YES	NO	YES, but only if the parameters are named similar syntactically to the rule criteria	NO
responseTime < 5	rt < 4	YES	YES	NO	NO
responseTime < 5	networkTime < 2 executionTime < 1	YES	NO	YES, but only if the parameters are named similar syntactically to the rule criteria	NO

The Matching Process



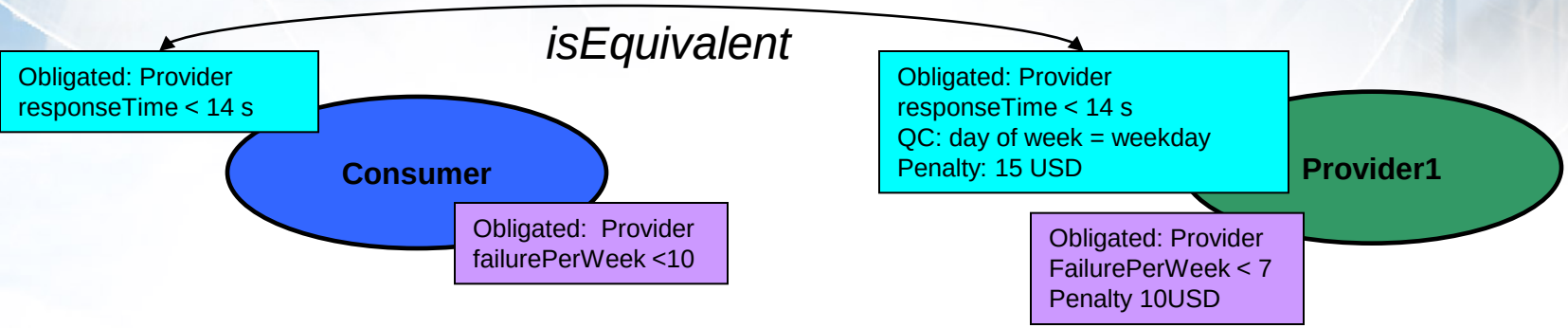
The Matching Process



Knowledge from Domain Specific Rules:

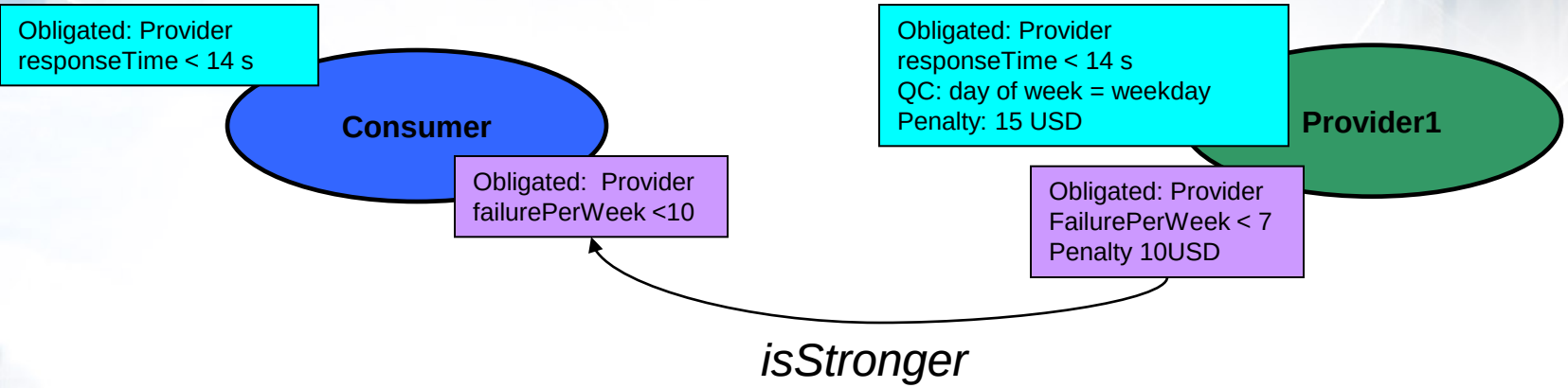
```
if (x >= 96)
  responseTime < y
else
  responseTime > y
```

The Matching Process



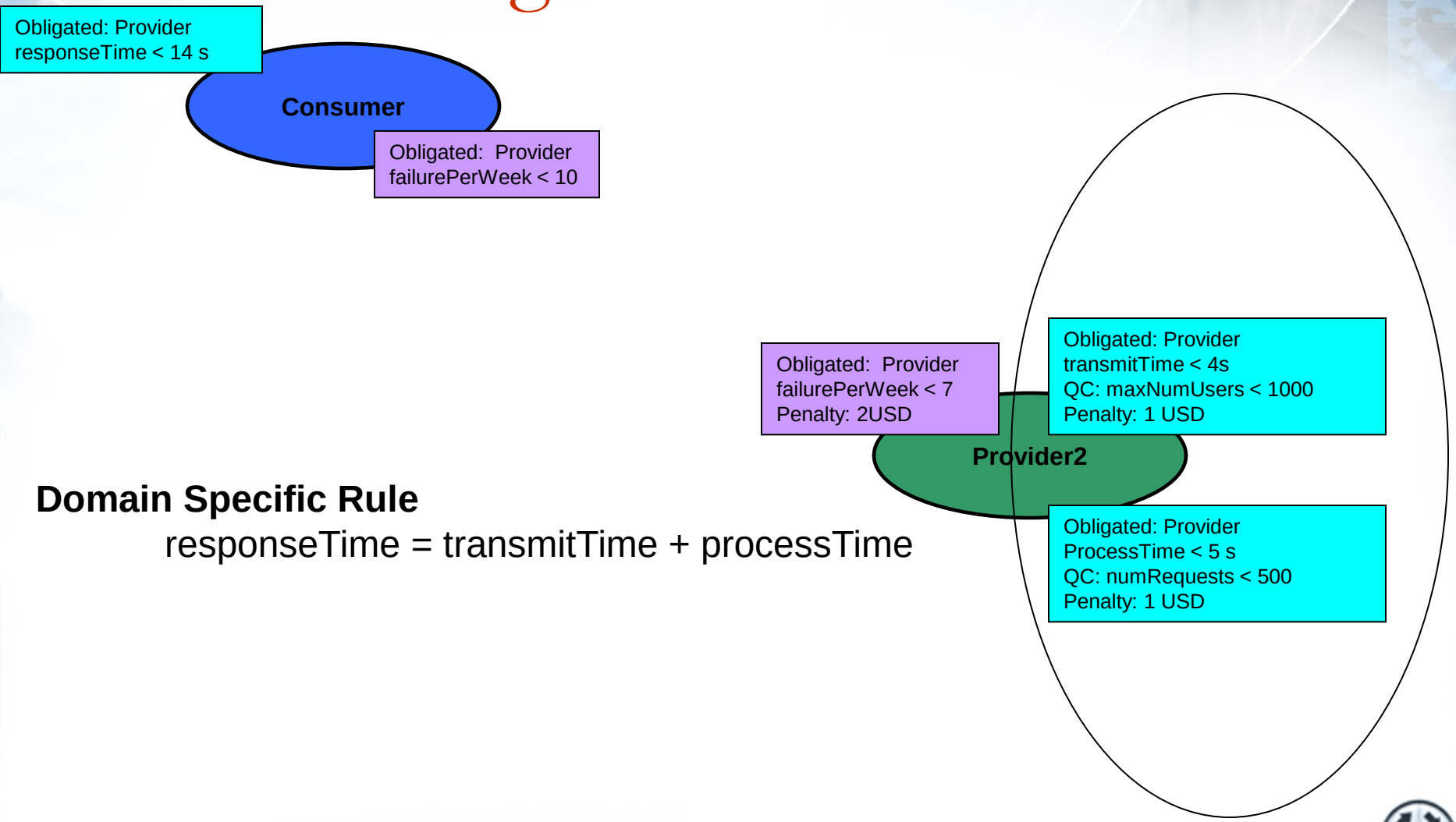
Knowledge from Semantics of Predicate Rules

The Matching Process



Knowledge from Semantics of Predicate Rules

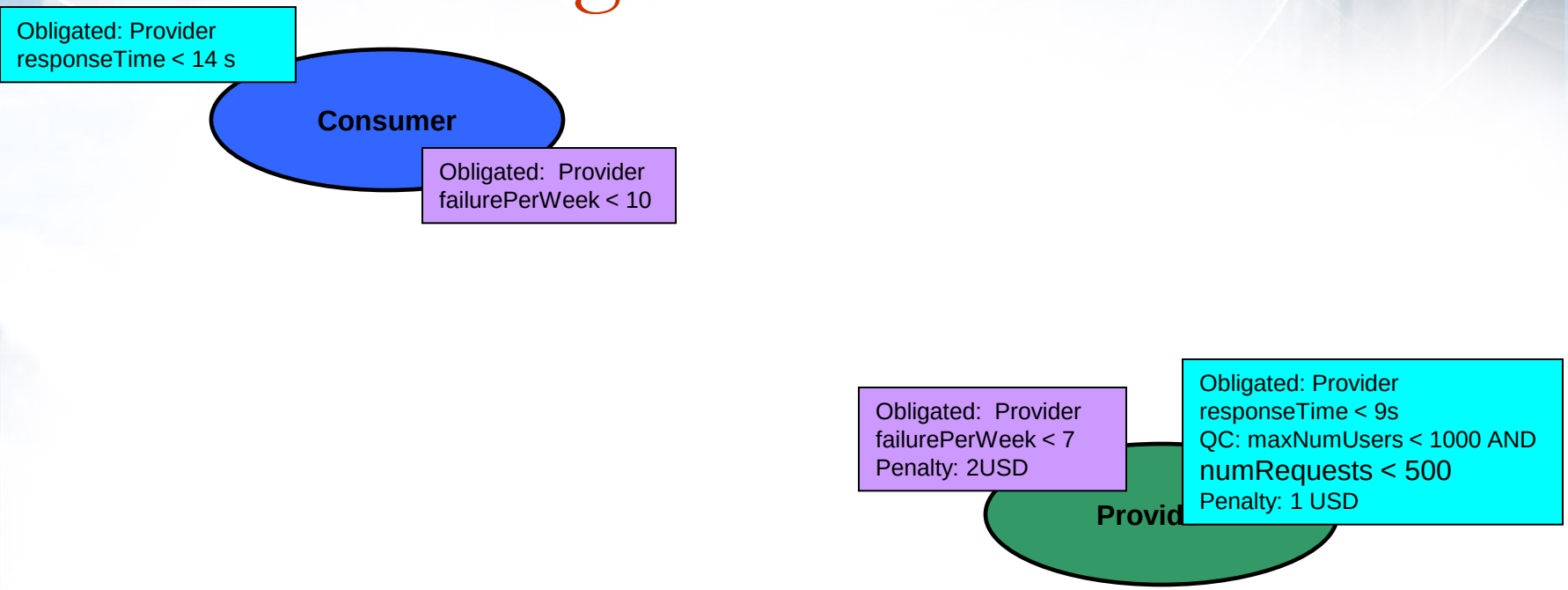
The Matching Process



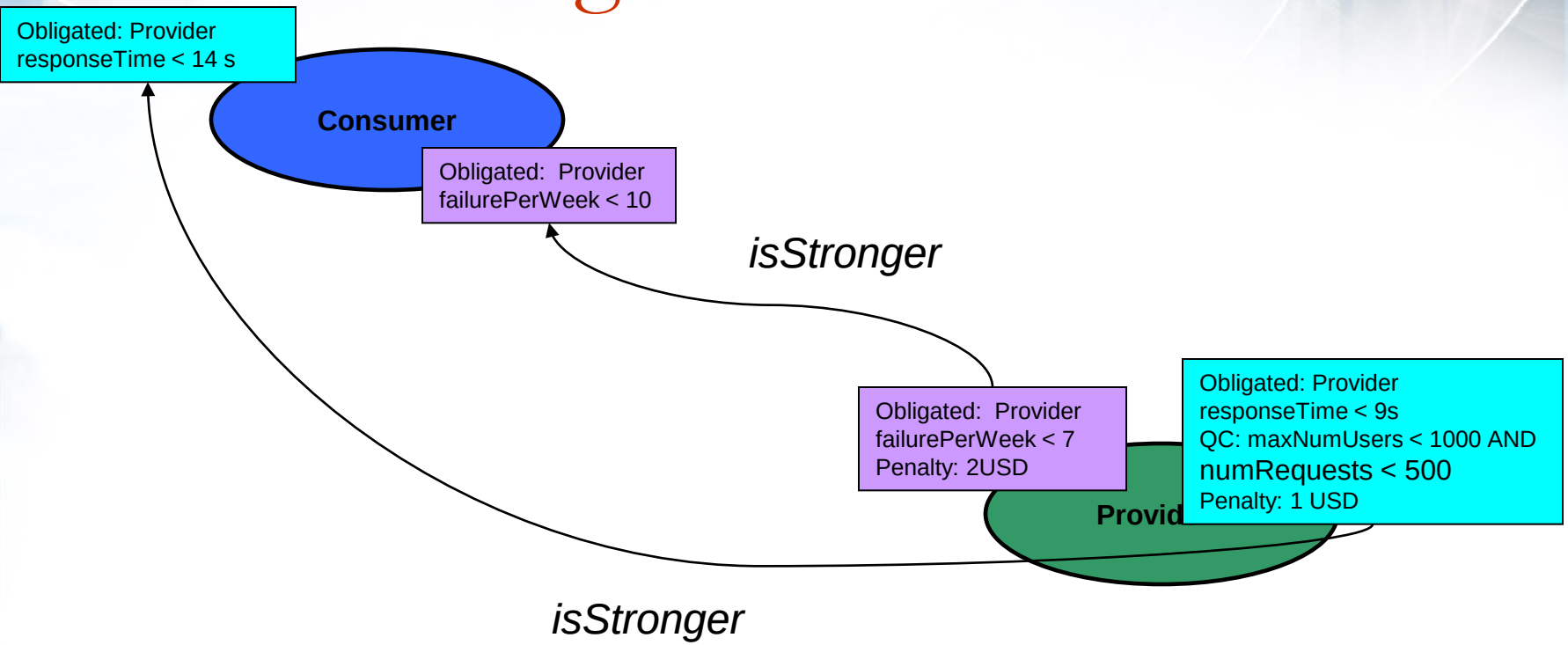
Domain Specific Rule

responseTime = transmitTime + processTime

The Matching Process



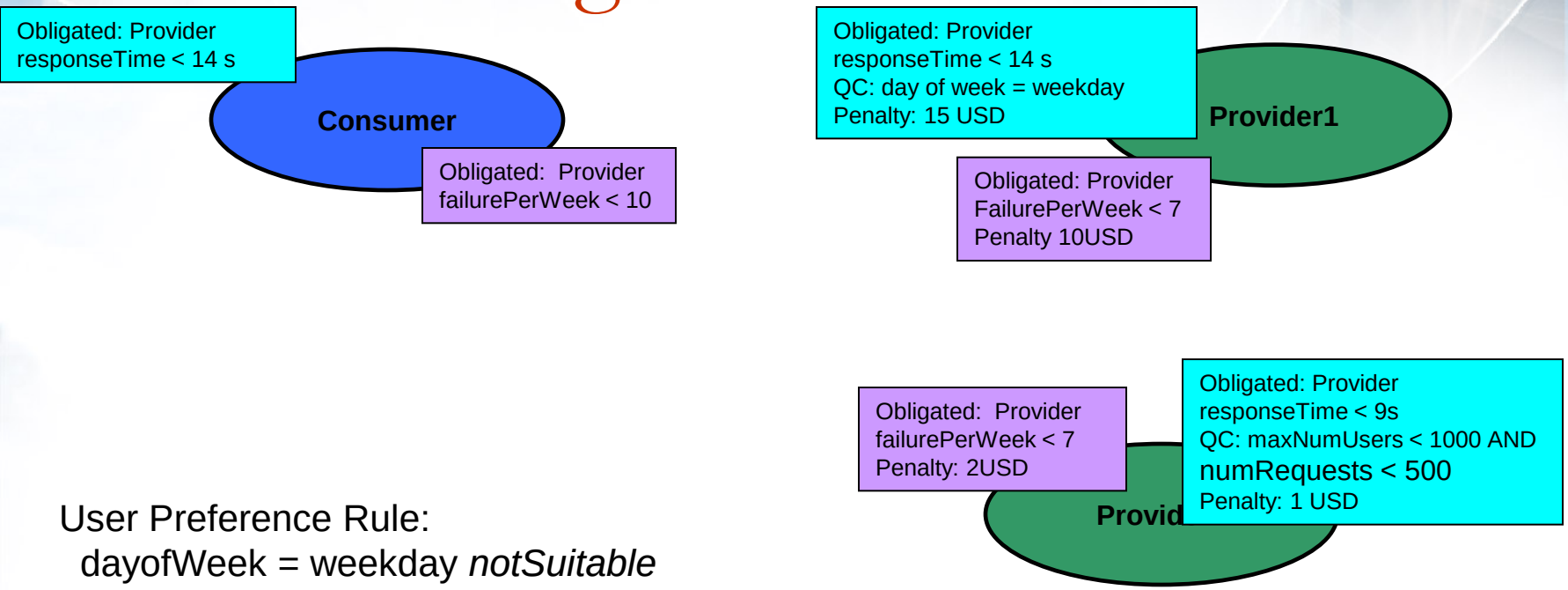
The Matching Process



Steps #5-6: Comparison Rules

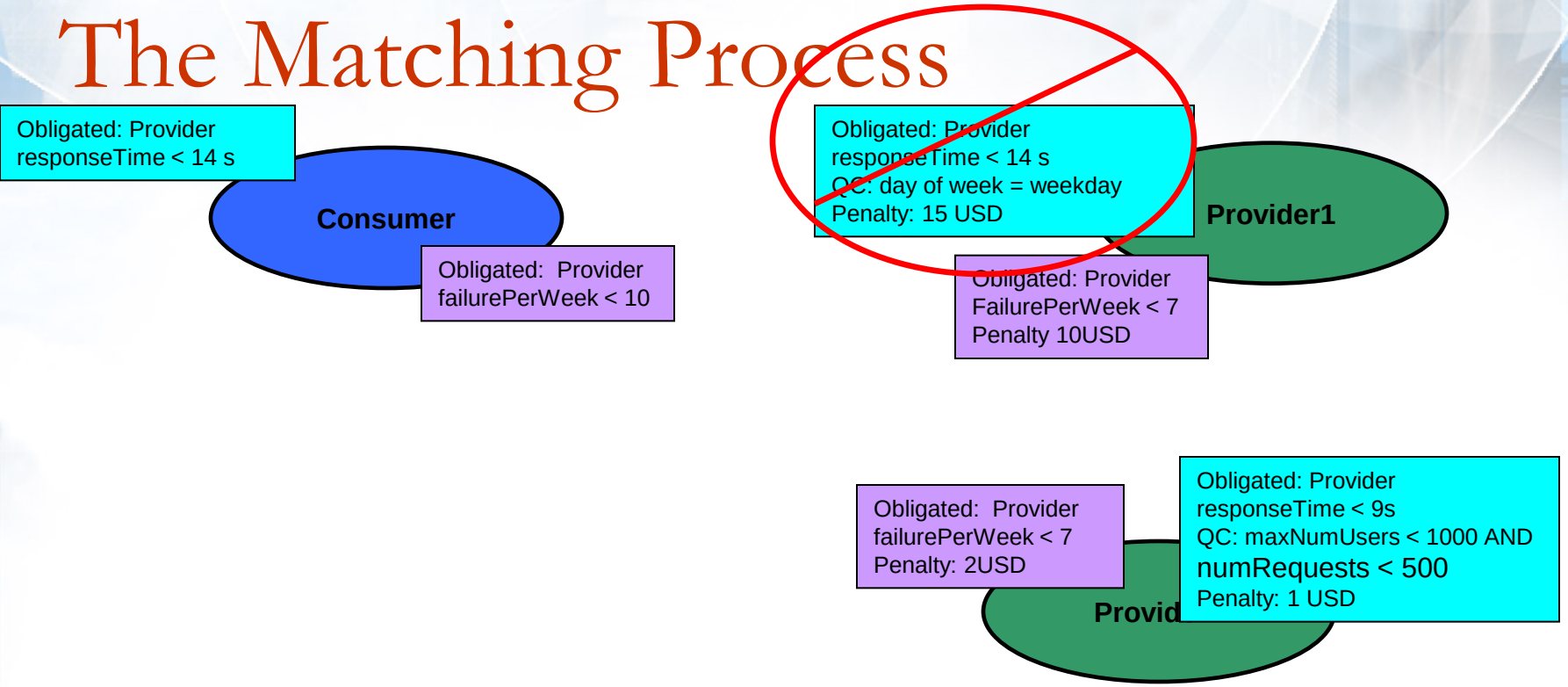
The Matching Process

notSuitable



User Preference Rule:
dayofWeek = weekday *notSuitable*

The Matching Process



Dynamic Process Configuration

- Operations Research has been used in industry for business process optimization
- There is often a lot of domain knowledge in business process optimization
  - Minds of analysts/experts
  - Hidden in databases/texts
- We try to explicitly capture domain knowledge and link with IT systems

Dynamic Process Configuration

Find optimal partners for the process based on process constraints – cost, supply time, etc.

Conceptual Approach

1. Create framework to capture represent domain knowledge
2. Represent constraints on the domain knowledge
3. Ability to reason on the constraints and configure the process

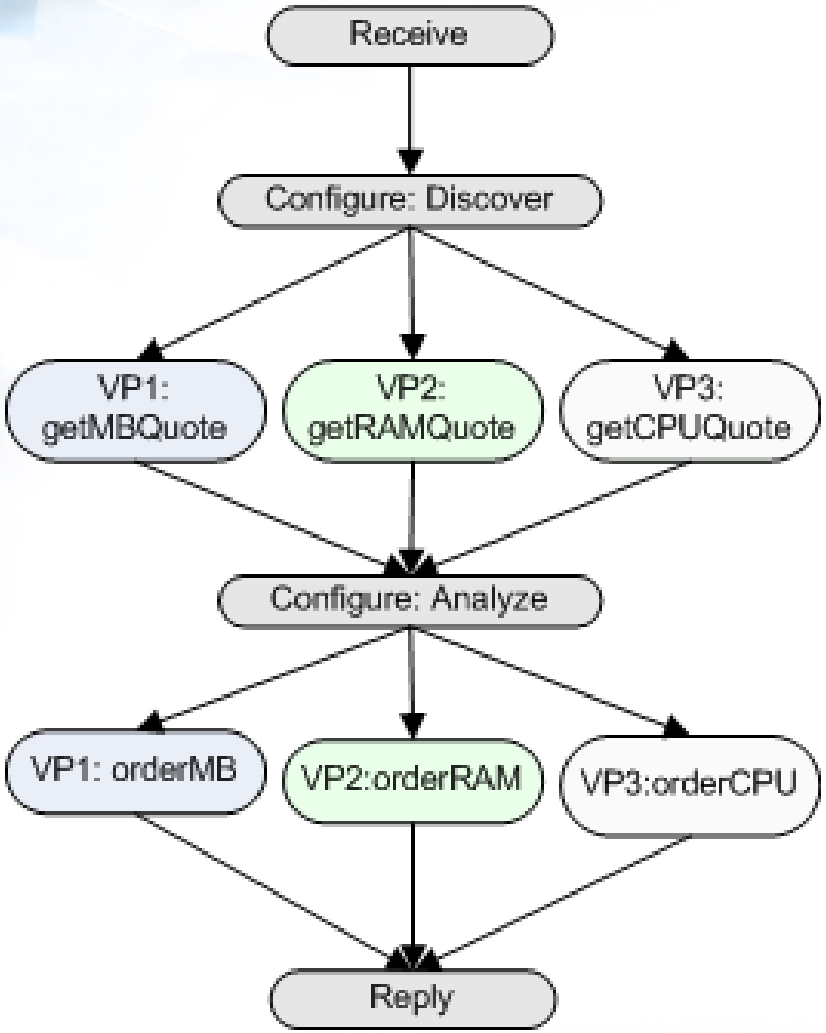
Dynamic Process Configuration

Research Challenges

- Capturing functional and non-functional requirements of the Web process (Abstract process specification)
- Discovering service partners based on functional requirements (Semantic Web service discovery)
- Choosing optimal partners that satisfy non-functional requirements (Constraint Analysis)

K. Verma, R. Akkiraju, R. Goodwin, P. Doshi, J. Lee, On Accommodating Inter Service Dependencies in Web Process Flow, AAAI Spring Symposium on Semantic Web Services, 2004
R. Aggarwal, K. Verma, J. A. Miller, Constraint Driven Composition in METEOR-S, SCC 2004.
K. Verma, K.Gomadam, J. Miller and A. Sheth, Configuration and Execution of Dynamic Web Processes, LSDIS Lab Technical Report, 2005.

Abstract Process Specification



1. Specify process control flow by using virtual partners
2. Specify Process Constraints
3. Capture Functional Requirements of Services using Semantic Templates

Process Constraints

- Constraints can be specified on a partner, an activity or the process as a whole.
- An objective function can also be specified e.g., minimize cost and supply-time, etc.
- Two types of constraints:
  - Quantitative (Q) (Time < 5 sec)
  - Logical (L) (preferredPartner, Security, etc.)

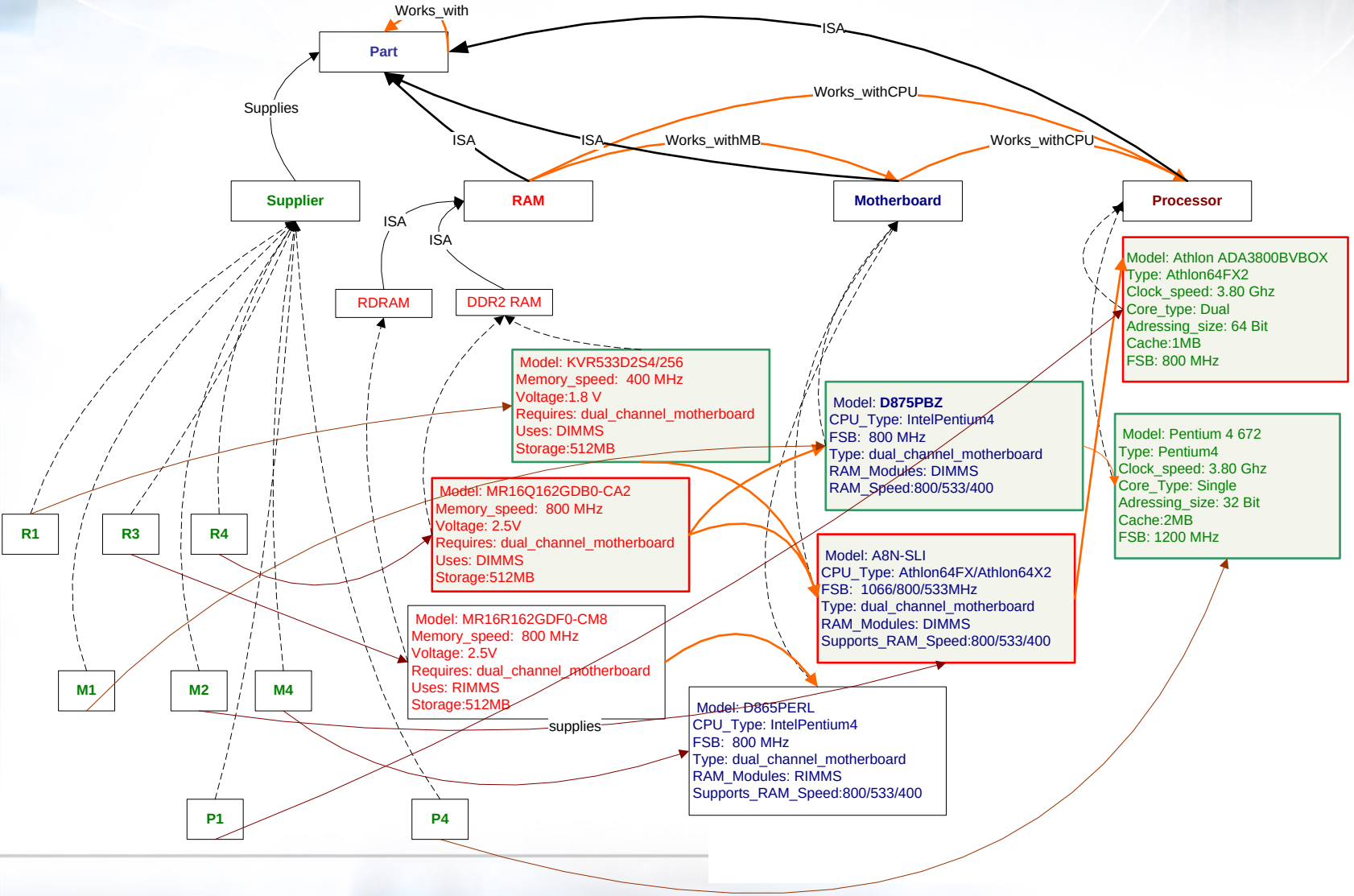
Process Constraints

Feature	Scope	Goal	Value	Unit	Aggregation
Cost (Quantitative)	Process	Minimize		Dollars	Σ
Supplytime (Quantitative)	Process	Satisfy	< 7	Days	MAX
Cost (Quantitative)	Activity	Satisfy	<200000	Dollars	Σ
PreferredSupplier(P1) (Logical)	Partner 1	Satisfy	True		
Compatible (P1, P2) (Logical)	Process	Satisfy	True		

Constraint Analysis

- Multi-paradigm proposed:
  - Integer Linear Programming for quantitative constraints
  - Semantic Web Rule Language and OWL for domain constraints
- Discovered Services first given to ILP solver
  - It returns ranked sets of services
- Then each set is checked for logical constraints using a SWRL reasoner
  - Sets not satisfying the criteria are rejected

Domain Ontology – Detailed View



Rules

- **Supplier 1 should be a preferred supplier.**
  - "if S1 is a supplier and its supplier status is preferred then the S1 is a preferred supplier".

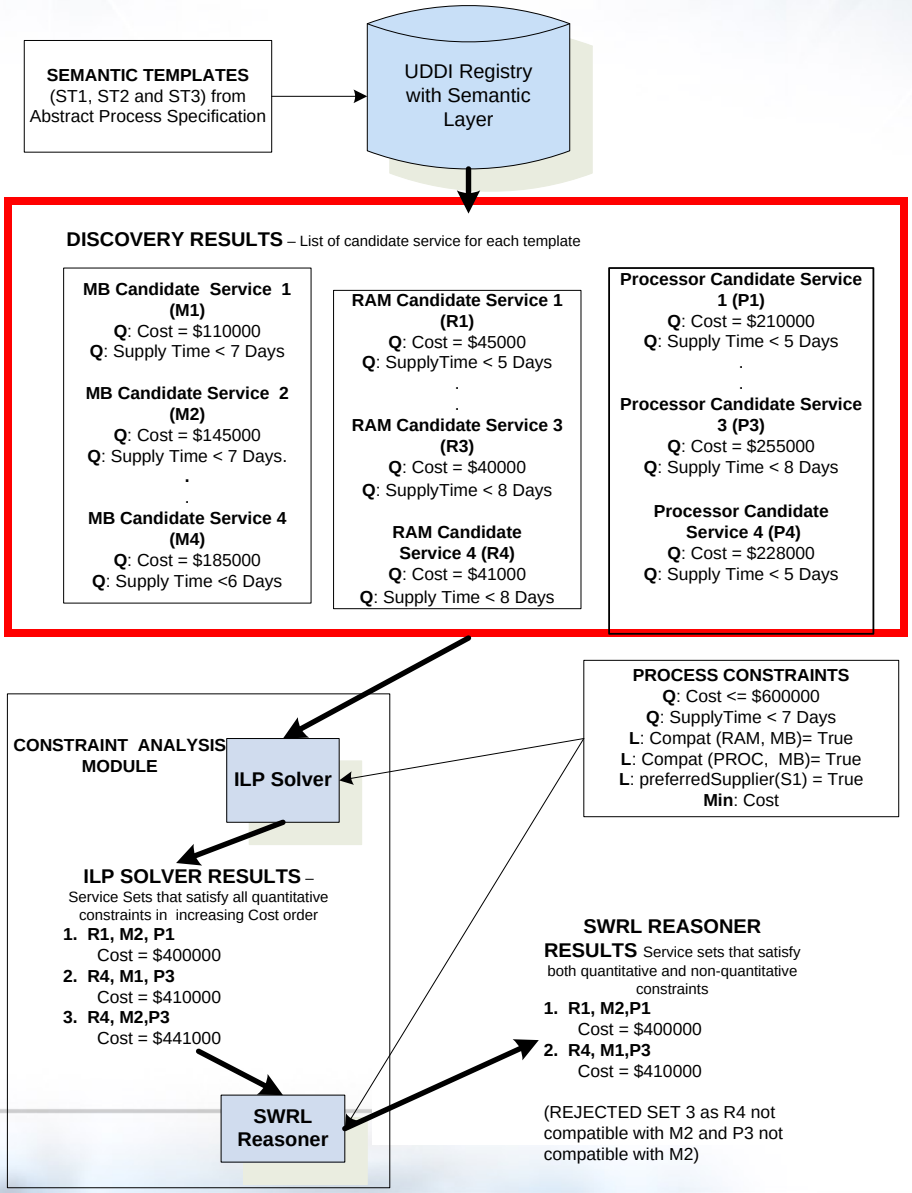
Supplier (?S1) and partnerStatus (?S1, "preferred") => preferredSupplier (?S1)

- **Supplier 1 and supplier 2 should be compatible.**
  - if S1 and S2 are suppliers and they supply parts P1 and P2, respectively, and the parts work with each other, then suppliers S1 and S2 are compatible for parts P1 and P2.

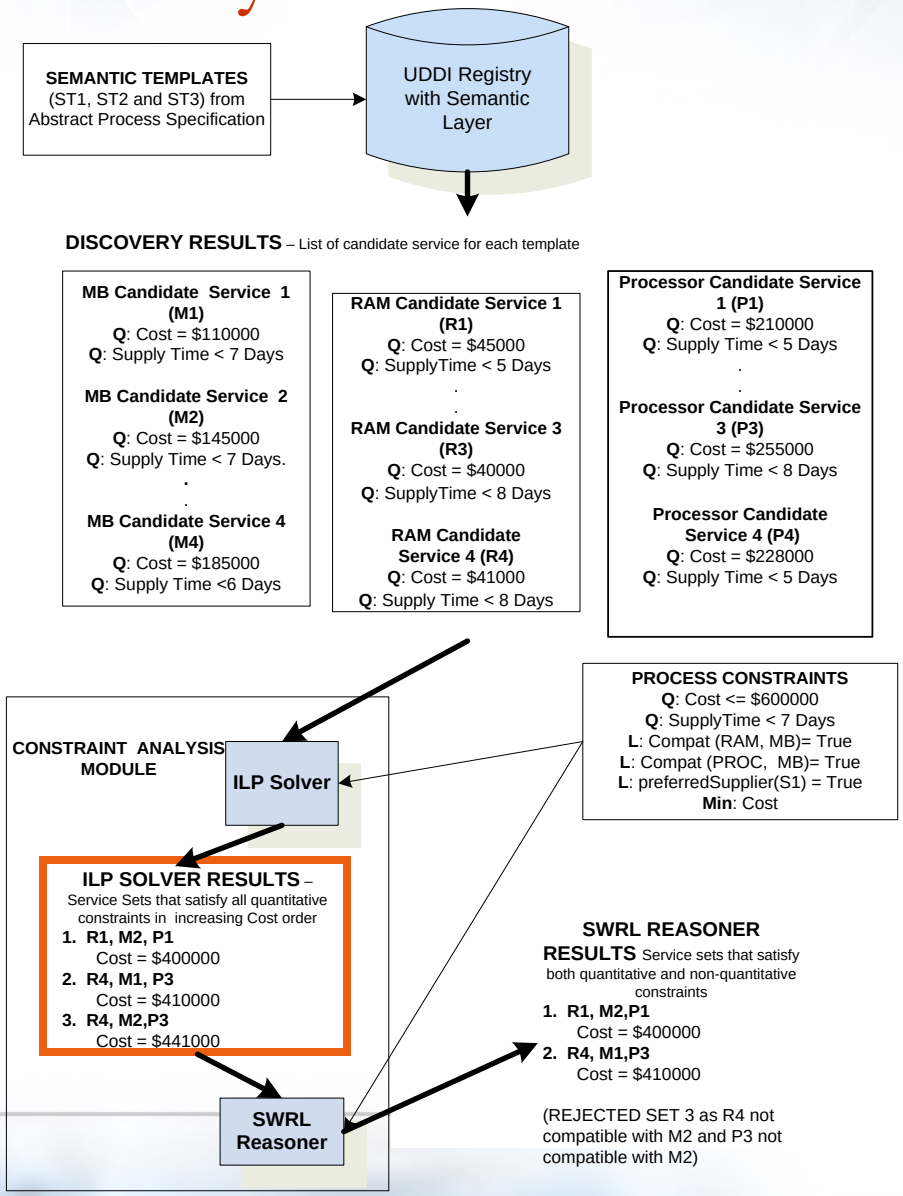
Supplier (?S1) and supplies (?S1, ?P1) and Supplier (?S2) and supplies (?S2, ?P2) and worksWith (?P1, ?P2) => compatible (?S1, ?S2, ?P1, ?P2)

RAM (?P1) and MB *(?P2) and worksWithMB (?P1, ?P2) =>worksWith (?P1, ?P2)*

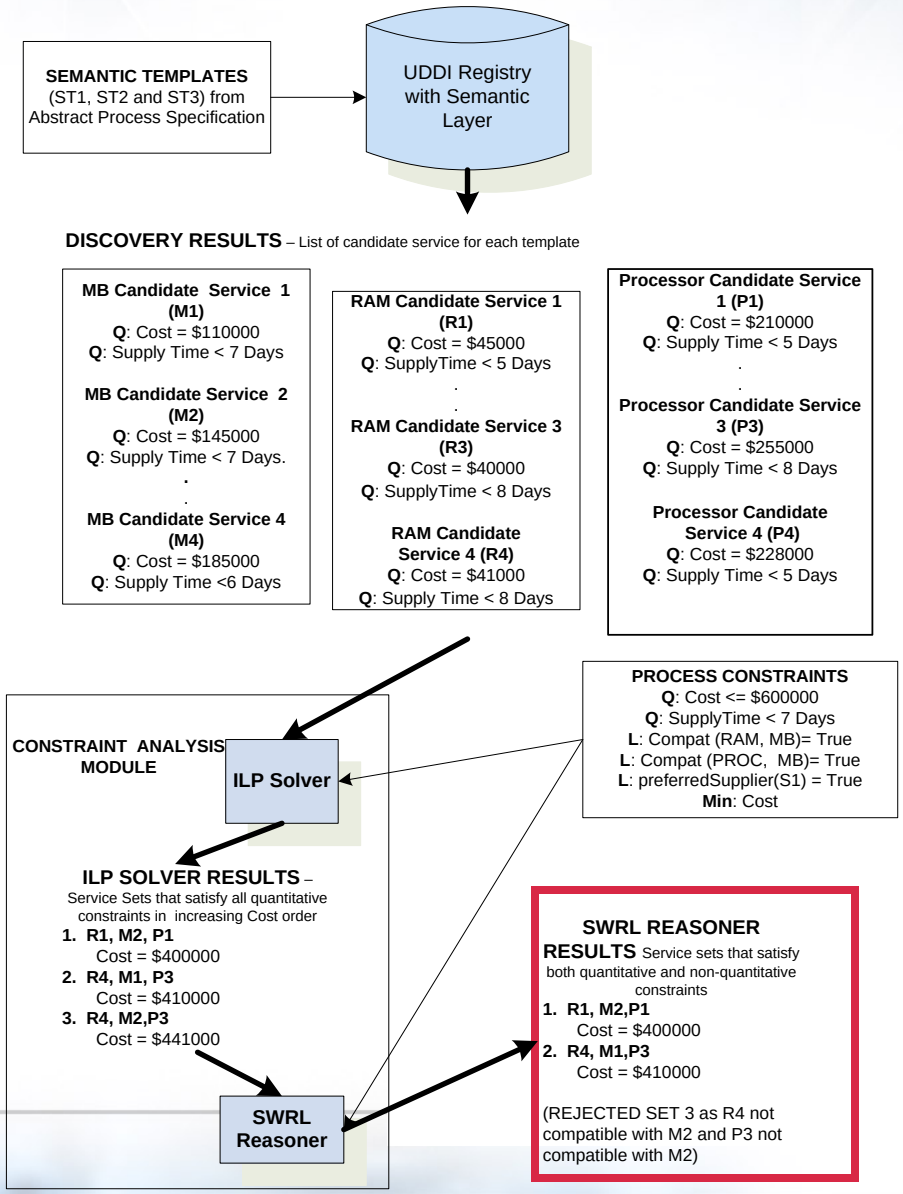
Configuration Step 1: Semantic Discovery



Configuration Step 2: Quantitative Constraint Analysis

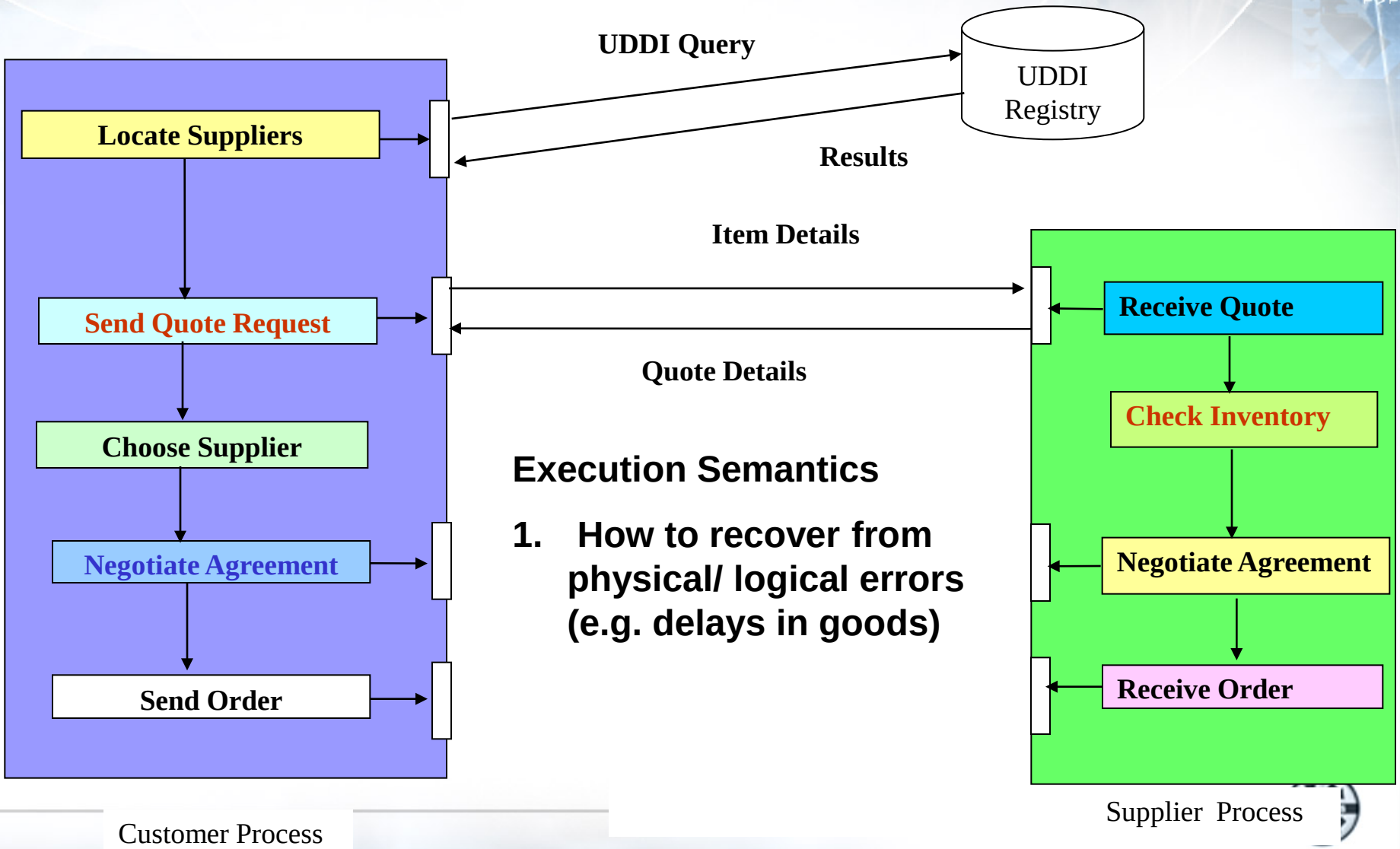


Configuration Step 3: Logical Constraint Analysis



EXECUTION SEMANTICS

Execution Semantics



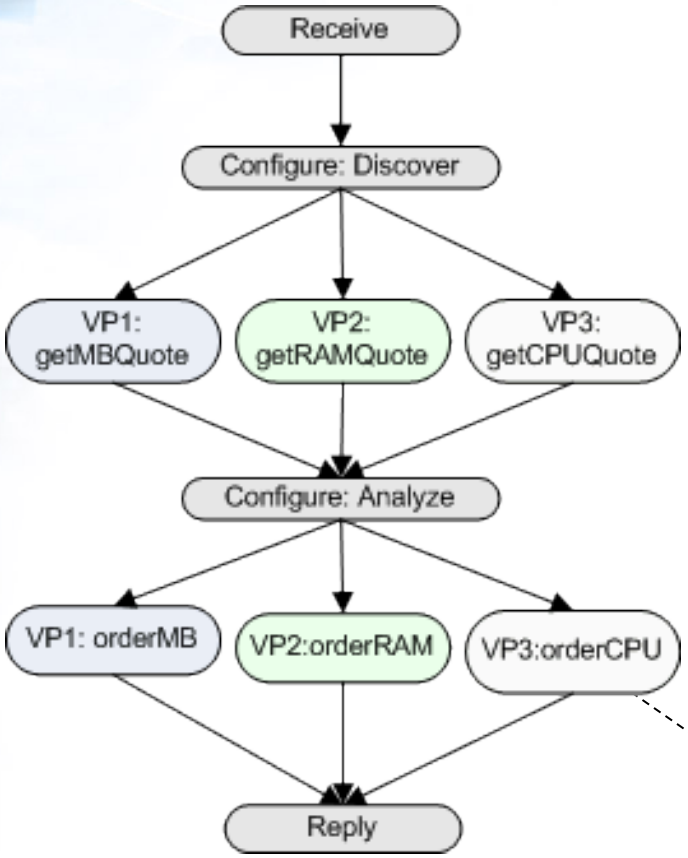
Process Adaptation

- Ability to adapt the processes from failures, unexpected events

- Two kinds of failures
  - Failures of physical components like services, processes, network
    - Can replace services using dynamic configuration
  - Logical failures like violation of SLA constraints/Agreements such as Delay in delivery, partial fulfillment of order
    - Need additional decision making capabilities

K. Verma, A. Sheth, Autonomic Web Processes, ICSOC 2005
K. Verma, P. Doshi, K. Gomadam, A. Sheth, J. Miller, Optimal Adaptation of Web Processes with Coordination Constraints, ICWS 2006.

Process Adaptation

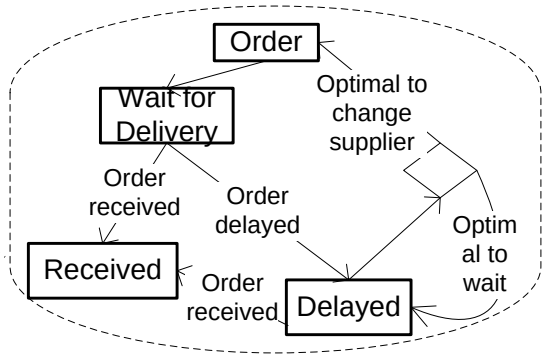


Adaptation Problem

Optimally react to events like delays in ordered goods

Conceptual Approach

1. Maintain states of the process – normal states, error states, goal states
2. Capture costs while transitioning from error states to goal state
3. Ability to decide optimal actions on the basis of state



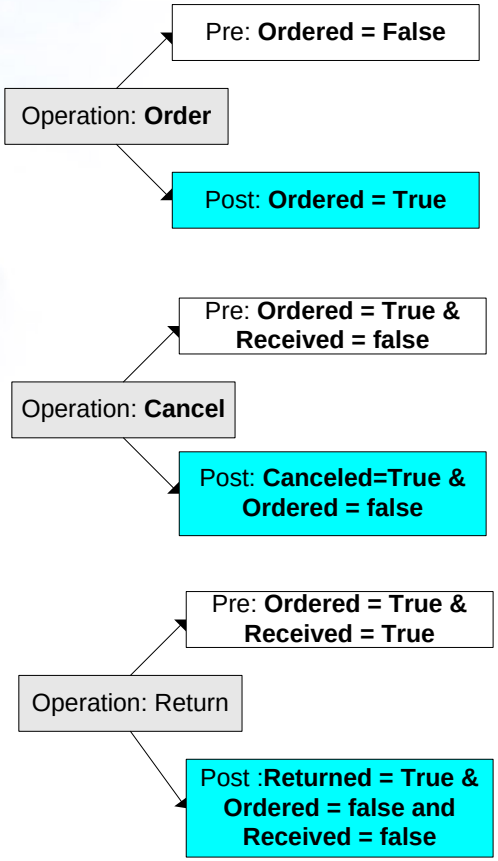
K. Verma, A. Sheth, Autonomic Web Processes, ICSOC 2005
K. Verma, P. Doshi, K. Gomadam, A. Sheth, J. Miller, Optimal Adaptation of Web Processes with Coordination Constraints, ICWS 2006.

Process Adaptation

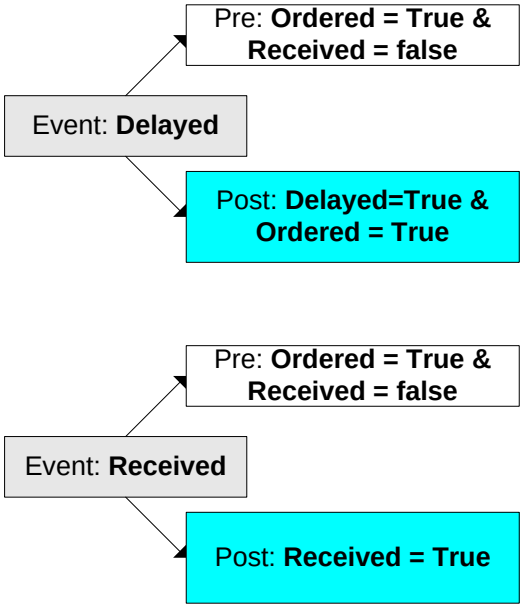
- Research Challenges
  - Creating a model to recover from failures and handle future events
  - Model must deal with two important factors
    - Uncertainty about when a failure occurs
    - Cost based recovery
- Scenario
  - After order for MB and RAM are placed, they may get delayed
  - The manufacturer may have severe costs if assembly is halted
  - It must evaluate whether it is cheaper to cancel/return and reorder or take the penalty of delay
  - Caveat: possible that reordered goods may be delayed too
- Proposed Solution
  - Modeling decision making capabilities of Service Managers as Markov Decision Processes (MDPs)

Generating States using preconditions and effects

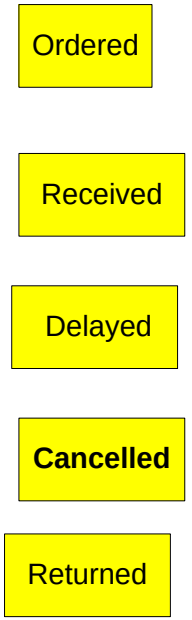
Actions



Events



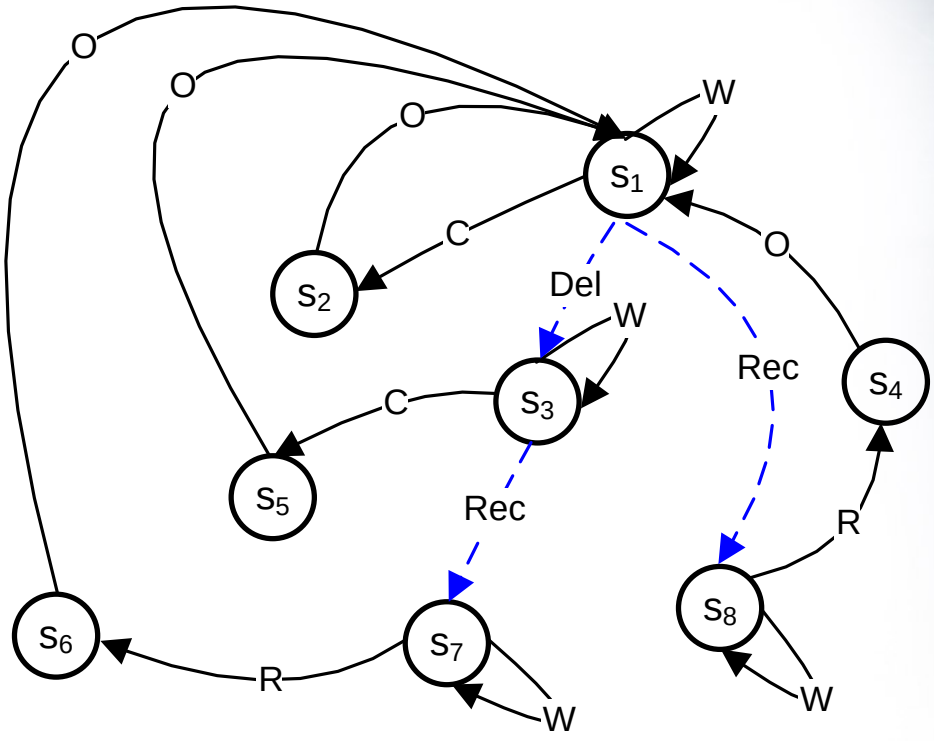
Flags



Use an algorithm similar to reachability analysis to generate states

Generated State Transition Diagram

State No.	Values of Boolean variables	Explanation
1	$<O\,\overline{C}\,\overline{R}\,\overline{Del}\,\overline{Rec}>$	Ordered
2	$<O\,C\,\overline{R}\,\overline{Del}\,\overline{Rec}>$	Ordered and Canceled
3	$<O\,\overline{C}\,\overline{R}\,Del\,\overline{Rec}>$	Ordered and Delayed
4	$<O\,\overline{C}\,R\,\overline{Del}\,Rec>$	Ordered, Received and Returned
5	$<O\,C\,\overline{R}\,Del\,\overline{Rec}>$	Ordered, Delayed and Cancelled
6	$<O\,\overline{C}\,R\,Del\,Rec>$	Ordered, Delayed, Received and Returned
7	$<O\,\overline{C}\,\overline{R}\,Del\,Rec>$	Ordered, Delayed and Received
8	$<O\,\overline{C}\,\overline{R}\,\overline{Del}\,Rec>$	Ordered and Received



Costs and Probabilities

- Costs of ordering taken from configuration module
  - From first two service sets
    - Optimal supplier and alternate supplier
- Probability of delay and cost of returning and canceling taken from supplier policy
  - Can be represented using WS-Policy or WS-Agreement

LSDIS

Large Scale Distributed Information Systems

University of Georgia
Computer Science Department

Semantics for Lightweight Services

Lightweight services and Mashups

- REST based implementation becoming popular
  - SOAP -> Web service
  - REST -> Lightweight Web service
- REST services exposed as API's
  - Eg. Google Maps API, Flickr API
- Mashups combine information from different services on the Web to create services with additional value
- Asynchronous Javascript And XML (AJAX) is primarily used by mashups to display the results to the user

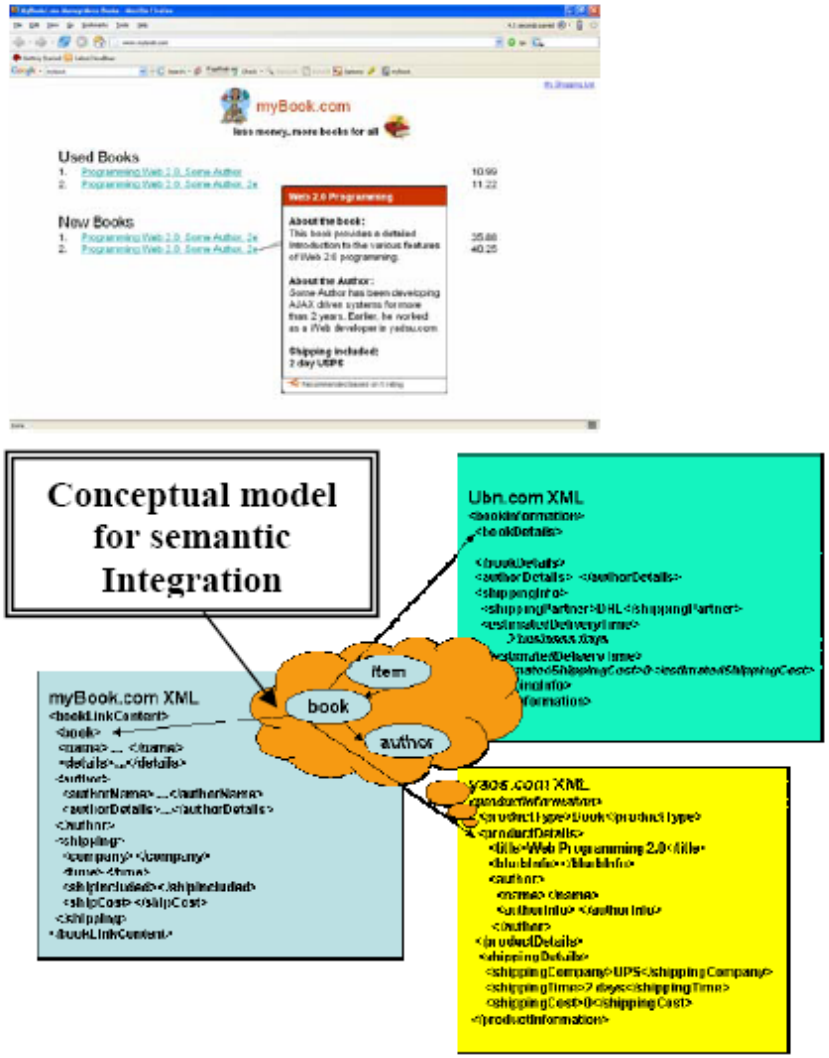
Current limitations and Role of semantics

- Current Mashups tightly coupled (lack dynamism)
  - E.g. HousingMaps.com uses craigslist and Google maps.
- Tight binding limits effectiveness
  - Better information may be available for a specific area
  - E.g. for Atlanta area, realtor1.com might be a better service than craigslist.
- Can annotate XML for automated integration

An example

- Consider a mashup: mybook.com
  - Allows users to search and buy used and new books
  - Gets data from various vendors on the web
- Can customize vendors based on requests
  - E.g., discover two vendors, ubn.com and yaos.com on the fly
- Use conceptual model/ontology based annotation of XML data for integration
  - mybook.com can interpret the XML documents from vendors with help of annotations

An Example of Smashup (Semantic mashup)



LSDIS

Large Scale Distributed Information Systems

University of Georgia
Computer Science Department

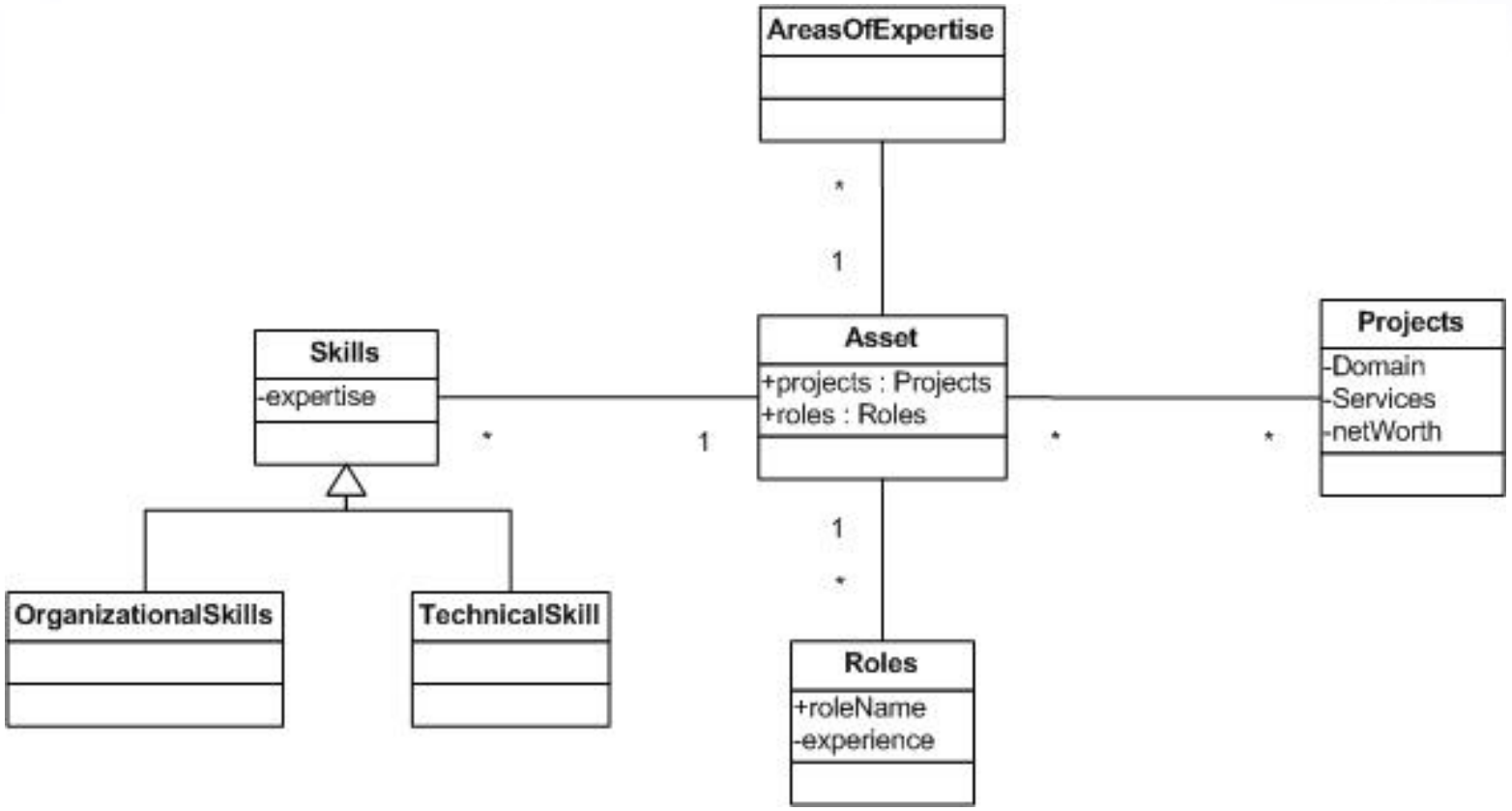
Semantics for Knowledge Services

Current and past focus of METEOR-S

Semantics for Knowledge Services

- Work in last two decades on knowledge modeling not so successful
  - Focus on capturing knowledge
  - However most businesses use people to solve problems not expert systems
- Knowledge service try to create semantic profiles of human expertise
  - Focus on “who can” not “how to”
  - Use of ontologies for shared descriptions

High Level Model for Knowledge Services



Using Model for Knowledge Services

- Such a model can be used to answer questions
  - Find managers who have led project worth at least a million dollars
  - Find developers who have created multimedia services using Java
  - Find consultants who have some expertise in Law

Autonomic Web Processes

- The goal (Albatross)
  - Self Configuring, Self Healing, Self Optimizing, Self Protecting Business Processes
- Realization
  - Comprehensive modeling of business processes using 3S model
- Advantages
  - Alignment of technology with business goals
  - Dynamic processes that adapt with the changing environment

Conclusions

- Businesses trying perceive IT as an extension of business strategy
  - 3S Model uses semantics to provide a comprehensive model of human and technical assets
  - Modeling and exploitation of four types of semantics
- CS Researchers must take a more pervasive view of services